

UNIVERSITÉ DU QUÉBEC À MONTRÉAL

MODÈLE MULTI-AGENTS POUR LE FILTRAGE
COLLABORATIF DE L'INFORMATION

THÈSE
PRÉSENTÉE
COMME EXIGENCE PARTIELLE
DU DOCTORAT EN INFORMATIQUE COGNITIVE

PAR
ZIED ZAIER

Janvier 2010

UNIVERSITÉ DU QUÉBEC À MONTRÉAL
Service des bibliothèques

Avertissement

La diffusion de cette thèse se fait dans le respect des droits de son auteur, qui a signé le formulaire *Autorisation de reproduire et de diffuser un travail de recherche de cycles supérieurs* (SDU-522 – Rév.01-2006). Cette autorisation stipule que «conformément à l'article 11 du Règlement no 8 des études de cycles supérieurs, [l'auteur] concède à l'Université du Québec à Montréal une licence non exclusive d'utilisation et de publication de la totalité ou d'une partie importante de [son] travail de recherche pour des fins pédagogiques et non commerciales. Plus précisément, [l'auteur] autorise l'Université du Québec à Montréal à reproduire, diffuser, prêter, distribuer ou vendre des copies de [son] travail de recherche à des fins non commerciales sur quelque support que ce soit, y compris l'Internet. Cette licence et cette autorisation n'entraînent pas une renonciation de [la] part [de l'auteur] à [ses] droits moraux ni à [ses] droits de propriété intellectuelle. Sauf entente contraire, [l'auteur] conserve la liberté de diffuser et de commercialiser ou non ce travail dont [il] possède un exemplaire.»

REMERCIEMENTS

Je tiens à remercier mes directeurs de thèse Robert Godin et Luc Faucher qui m'ont offert la possibilité de réaliser ce travail et m'ont encadré toutes ces années.

Je remercie particulièrement Esma Aïmeur, Daniel Lemire, Daniel Memmi, Petko Valtchev et Pierre Poirier pour les nombreux commentaires et conseils portant sur ce travail.

Je voudrais adresser un remerciement à tous les membres du DIC, en particulier, Mounir Boukadoum et Ghislain Lévesque ainsi que Marc Bouisset, Wilfried Probst et Jude Nsiempba; et aux membres du LATECE, en particulier, Hafedh Mili.

Pour finir, je remercie mes parents, Fourat et Fares, Elham, Houda, Nadéra et Fred ainsi que tout le reste de ma famille.

TABLE DES MATIÈRES

LISTE DES FIGURES	vi
LISTE DES TABLEAUX	ix
RÉSUMÉ	x
ABSTRACT	xi
CHAPITRE I INTRODUCTION.....	1
1.1. MOTIVATION	1
1.2. PROBLÈMES DE RECHERCHE	3
1.3. PROPOSITION DE RECHERCHE.....	8
1.3.1. Approche proposée	8
1.3.2. Contributions	9
1.3.3. Travaux publiés	11
1.4. PRÉSENTATION DE LA RECHERCHE.....	12
CHAPITRE II FILTRAGE DE L'INFORMATION	15
2.1. INTRODUCTION.....	15
2.2. COLLECTE DES PRÉFÉRENCES.....	17
2.3. GRANDES FAMILLES DE FILTRAGE	21
2.3.1. Filtrage basé sur le contenu	22
2.3.2. Filtrage collaboratif	25
2.3.3. Filtrage hybride.....	29
2.4. CONCLUSION	30
CHAPITRE III SYSTÈME DE RECOMMANDATION DISTRIBUÉ.....	32
3.1. MOTIVATION	32
3.2. DOMAINES D'APPLICATION DES RÉSEAUX PAIR-À-PAIR	33
3.3. CLASSES DE RÉSEAUX PAIR-À-PAIR	35
3.4. TECHNIQUES DE REPÉRAGE DE L'INFORMATION POUR LES RÉSEAUX PAIR-À-PAIR	39

3.4.1. Approches centralisées	39
3.4.2. Approches décentralisées non structurées	39
3.4.2.1. Techniques de recherche à l'aveugle.....	40
3.4.2.2. Techniques de recherche dirigée	43
3.4.3. Approches décentralisées structurées	45
3.5. SYSTÈMES DE RECOMMANDATION DISTRIBUÉS	49
3.6. CONCLUSION	51
CHAPITRE IV AGENT SOCIAL DE RECOMMANDATION	52
4.1. INTRODUCTION.....	52
4.2. INTELLIGENCE ARTIFICIELLE DISTRIBUÉE ET SYSTÈMES MULTI-AGENTS	
54	
4.2.1. Intelligence artificielle distribuée	54
4.2.2. Agents et systèmes multi-agents.....	56
4.2.3. Formes de l'intelligence artificielle distribuée	58
4.3. INTÉRÊT DES COMPORTEMENTS SOCIAUX COMPLEXES DANS LES	
SYSTÈMES MULTI-AGENTS.....	61
4.4. PHÉNOMÈNES SOCIAUX	66
4.5. CONCLUSION	73
CHAPITRE V PROPOSITION DE RECHERCHE.....	76
5.1. INTRODUCTION.....	76
5.2. MÉTHODE DE FILTRAGE.....	77
5.2.1. Méthode d'évaluation des documents.....	79
5.2.2. Représentation adoptée pour le profil thématique	80
5.2.3. Discrimination du voisinage	81
5.2.4. Calcul de la similarité entre utilisateurs.....	82
5.2.5. Calcul des prédictions.....	82
5.3. ARCHITECTURE DU SYSTÈME.....	83
5.4. PLATEFORME MULTI-AGENT	86

LISTE DES FIGURES

Figure 1.1 Système de recommandation centralisé.....	5
Figure 1.2 Magellan.....	6
Figure 1.3 Explication de la recommandation d' <i>Amazon</i> (Vellino et Zeber, 2008)	7
Figure 2.1 Modèle général pour le filtrage d'information, inspiré de Berrut et Denos (2003).....	17
Figure 2.2 Aspects de vie quotidienne pouvant être utilisés par les systèmes de recommandation (Miller, 2003)	21
Figure 2.3 Modèle général pour le filtrage collaboratif de l'information (Berrut et Denos, 2003).....	26
Figure 3.1 Topologie du réseau <i>Napster</i>	36
Figure 3.2 Topologie du réseau <i>Kazaa</i>	37
Figure 3.3 Représentation symbolique des réseaux pair-à-pair non structurés.....	37
Figure 3.4 Représentation symbolique du DHT	38
Figure 3.5 Table de hachage distribuée	39
Figure 3.6 Approche centralisée	40
Figure 3.7 Technique <i>BFS</i> (Zeinalipour-Yazti, Kalogeraki et Gunopulos, 2004)	41
Figure 3.8 Technique <i>RBFS</i> (Zeinalipour-Yazti, Kalogeraki et Gunopulos, 2004).....	42
Figure 3.9 Technique « <i>Random Walkers</i> » (Zeinalipour-Yazti, Kalogeraki et Gunopulos, 2004)	43
Figure 3.10 Technique <i>>RES</i> (Zeinalipour-Yazti, Kalogeraki et Gunopulos, 2004).....	44

Figure 3.11 Recherche en profondeur (Zeinalipour-Yazti, Kalogeraki et Gunopulos, 2004)	45
Figure 3.12 <i>Chord</i>	46
Figure 3.13 Table des pointeurs	46
Figure 3.14 <i>CAN</i>	47
Figure 3.15 <i>Tapestry</i>	48
Figure 3.16 Technique <i>ISM</i> (Zeinalipour-Yazti, Kalogeraki et Gunopulos, 2004)	48
Figure 4.1 Les trois axes fondamentaux de recherche de l'intelligence artificielle distribuée	55
Figure 4.2 Agent d'après (Russell et Norvig, 2004)	57
Figure 4.3 Pointillisme	62
Figure 4.4 Les niveaux hiérarchiques des organismes d'après Vaario (1994)	63
Figure 4.5 Un exemple de comportements d'agrégation (De Waal, 2000)	68
Figure 4.6 Un exemple de comportements de coopération (Kappeler, 2005)	69
Figure 4.7 Un première exemple de tromperie sociale d'après (Byrne, 1995)	70
Figure 4.8 Un deuxième exemple de tromperie sociale d'après (Byrne, 1995)	71
Figure 4.9 Un troisième exemple de tromperie sociale d'après (Byrne, 1995)	72
Figure 5.1 Technique de discrimination du voisinage	81
Figure 5.2 Architecture générale d'un système de recommandation centralisé	83
Figure 5.3 Architecture générale de systèmes décentralisés de recommandation	84
Figure 6.1 La courbe <i>ROC</i> (Herlocker <i>et al.</i> , 2004)	97

Figure 6.2 Filtrage collaboratif	101
Figure 6.3 Filtrage hybride	102
Figure 6.4 Prétraitement : Téléchargement du contenu	102
Figure 6.5 Prétraitement : Création du profil du document et de l'utilisateur	103
Figure 6.6 Prétraitement : Création du profil utilisateur démographique	103
Figure 6.7 Filtrage hybride démographique.....	104
Figure 6.8 Protocole d'évaluation.....	105
Figure 6.9 MAE pour le filtrage collaboratif en fonction de la taille du voisinage	106
Figure 6.10 MAE pour le filtrage hybride en fonction de la taille du voisinage	107
Figure 6.11 MAE pour le filtrage hybride démographique en fonction de la taille du voisinage	107
Figure 6.12 Rappel pour le filtrage collaboratif en fonction de la taille du voisinage....	108
Figure 6.13 Rappel pour le filtrage hybride en fonction de la taille du voisinage	109
Figure 6.14 Rappel pour le filtrage hybride démographique en fonction de la taille du voisinage	109
Figure 6.15 Couverture pour le filtrage collaboratif en fonction de la taille du voisinage	110
Figure 6.16 Couverture pour le filtrage hybride en fonction de la taille du voisinage ...	111
Figure 6.17 Couverture pour le filtrage hybride démographique en fonction de la taille du voisinage	111

LISTE DES TABLEAUX

Tableau 2.1 Recherche d'informations versus filtrage d'informations (Berrut et Denos, 2003).....	16
Tableau 2.2 Une matrice des notes attribuées aux articles par les utilisateurs.....	27
Tableau 2.3 Méthodes d'hybridation (Burke, 2002)	30
Tableau 4.1 Les différences entre les agents cognitifs et réactifs	61
Tableau 5.1 Filtrage collaboratif basé sur les notes	78
Tableau 5.2 Filtrage collaboratif basé sur le contenu	78
Tableau 6.1 Les classes de documents.....	95

RÉSUMÉ

Les systèmes de recommandation sont nés de la volonté de pallier le problème de surcharge d'information du web. Combinant des techniques de filtrage d'information, personnalisation, intelligence artificielle, réseaux sociaux et interaction personne-machine, les systèmes de recommandation fournissent à des utilisateurs des suggestions qui répondent à leurs besoins et préférences informationnelles. En effet, les systèmes de recommandation sont particulièrement sollicités dans les applications de commerce électronique.

Cependant, ce type de système a été en grande partie confiné à une architecture centralisée. Récemment, l'architecture distribuée a connu une popularité croissante, comme en témoigne par exemple, les réseaux pair-à-pair (« *peer-to-peer* »), le calcul distribué (« *Grid computing* »), le web sémantique, etc., et s'impose peu à peu comme une alternative à l'approche client/serveur classique. L'hypothèse des chercheurs est que les systèmes de recommandation peuvent tirer profit d'une architecture distribuée.

Dans cette thèse, nous étudions les défis que posent les systèmes de recommandation distribués et nous proposons une nouvelle architecture pair-à-pair, de filtrage collaboratif, basée sur la discrimination du voisinage. Nous étudions l'évolution de la performance, de la couverture et de la qualité des prédictions pour différentes techniques de recommandation. En outre, nous identifions la méthode de recommandation la plus efficace pour cette nouvelle architecture pair-à-pair.

Bien que cette thèse se concentre essentiellement sur le domaine décentralisé de système de recommandation, nos contributions ne se limitent pas strictement à ce domaine de recherche. En effet, ces contributions touchent des problèmes de recherche dans plusieurs autres domaines de recherche (système multi-agents, gestion des profils utilisateurs, réduction de la complexité computationnelle, collecte des préférences utilisateurs, PageRank, etc.).

Mots clés : Filtrage de l'information, Filtrage collaboratif, système de recommandation, système distribué, agent social.

ABSTRACT

Recommender systems are considered an answer to the information overload in a web environment. Combining ideas and techniques from information filtering, personalization, artificial intelligence, user interface design and human-computer interaction, recommender systems provide users with proactive suggestions that are tailored to meet their particular information needs and preferences. Indeed recommender system plays an essential role in e-commerce applications.

However, this type of system has been largely confined to a centralized architecture. Recently, distributed architectures are becoming more and more popular (as witnessed by peer-to-peer, Grid computing, semantic web, etc.) and try replacing classical client/server approach. Recommender system could likewise profit from this architecture. Indeed, novel decentralized recommender systems are emerging.

In this thesis, we investigate the challenges that decentralized recommender systems bring up and propose a new peer-to-peer collaborative filtering architecture based on prior selection of the neighbors. We investigate the evolution of different recommendation techniques performance, coverage and quality of prediction. Also, we identify which recommendation method would be the most efficient for this new peer-to-peer architecture.

While this thesis mainly concentrates on decentralized collaborative filtering recommender system domain, our contributions are not only confined to this research domain. Indeed, many of these contributions address issues relevant to other research domains (multi-agent systems, user profile management, computational complexity reduction, collecting preference information, PageRank, etc.).

Keywords : Information filtering, Collaborative filtering, Recommender system, Distributed system, Social agent.

Chapitre I

Introduction

« Trop d'information tue l'information »

- Anonyme

1.1. Motivation

Le réseau Internet connaît depuis quelques années un accroissement très important du nombre d'utilisateurs, du nombre d'ordinateurs connectés et de la quantité d'information qui y est disponible (Miller, Konstan et Riedl, 2004). En effet, selon *Computer Industry Almanac*, le nombre total d'internautes, en 2005, dépasserait le milliard, soit deux fois plus qu'en 2001 (Computer Industry Almanac, 2006). De plus, en 2006, on a estimé à 161 Exaoctets les données mises en ligne depuis le dévoilement du web (Gantz *et al.*, 2007), contre 7500 Téraoctets en 2001 (Miller, Konstan et Riedl, 2004).

La nature même de ce réseau en a fait, dès le début, un outil potentiel pour la recherche d'informations. L'échange d'information est potentiellement plus rapide, plus facile et moins coûteux (courriel, clavardage, vidéoconférence, etc.) que les moyens traditionnels. De même, le réseau offre un accès instantané à une quantité importante d'information variée grâce au nombre considérable de blogues, forums, médias sociaux, réseaux sociaux, etc. Par ailleurs, l'accès à l'information par l'intermédiaire du réseau n'est plus limité aux informations publiques et gratuites : les encyclopédies et bases de données commerciales proposent de plus en plus des accès à leurs sources d'informations via Internet (*Encarta*, *Universalis*, etc.). De même, les journaux offrent un accès à leurs archives gratuitement ou de manière payante (Malaisson, 2007).

Internet est donc en train de s'imposer comme source d'information complémentaire aux services d'information classiques. Cependant, il est difficile de trouver les données que l'on cherche dans cette montagne d'information. Il faut pouvoir retrouver l'information pertinente sur le réseau et savoir l'exploiter. Or, la nature même de l'Internet veut qu'elle soit dispersée, non centralisée. De plus, la liberté complète de publication sur le web a comme inconvénient l'apparition de nombreuses sources d'informations non pertinentes (sites web non remis à jour, désinformations, publicités, etc.). De même, l'information disponible sur le réseau est dynamique. Ainsi, un des enjeux est d'identifier les sites qui sont fréquemment mis à jour et la dernière fois qu'ils l'ont été. Enfin, malgré la présence de champs pour les métadonnées, nommés *Balises*, dans le langage *HTML*, les données proposées sur le web sont, dans la majorité des cas, faiblement structurées (Kim, Kim et Cho, 2008 ; Miller, 2003 ; Ziegler, 2005).

Ces contraintes ont transformé la recherche d'informations sur Internet en une démarche ardue. La quête des ressources pertinentes sur un sujet donné est donc logiquement devenue un problème essentiel de recherche pour les internautes. Certains des sites les plus visités sont ceux qui ont pour objectif d'aider les utilisateurs dans leurs recherches au moyen de répertoires (*Yahoo*, *Magellan*, etc.) ou au moyen de moteurs de recherche (*Google*, *Altavista*, *HotBot*, etc.). Effectivement, d'après *Search Engine Watch*, les outils de recherche d'informations sont les applications les plus répandues sur le Web. De plus, on a évalué à plus de 213 millions le nombre de recherches effectuées chaque jour (Sullivan, 2006). Mais devant l'extraordinaire croissance et diversité des ressources disponibles sur le réseau, ces outils se révèlent de plus en plus insuffisants. Selon de récentes études, 93 % de l'information produite dans le monde est sous forme numérique (Kim, Kim et Cho, 2008). De plus, on a recensé, en 2007, un peu moins de 120 millions de sites web, contre 6 millions en 1999 (Netcraft, 2007).

Les internautes se trouvent confrontés à un très grand nombre de réponses non pertinentes retournées par les outils de recherche d'informations classiques. Ceci est dû aux insuffisances des systèmes existants de recherche d'informations. Par exemple, la simple présence du terme recherché dans un document suffit en effet, dans la logique de ces outils de recherche, à le rendre potentiellement pertinent. Cependant, un terme peut être employé

dans de multiples contextes et pour de nombreuses raisons sans que le document qui le contient soit pour autant pertinent pour l'utilisateur. Par exemple, il existe le problème de la synonymie : une simple requête sur l'« *automobile* » laissera de côté un grand nombre de pages qui évoquent le terme « *voiture* » ; la polysémie : une recherche sur « *pêche* » donnera des documents concernant, entre autres, le fruit et les activités de pêche ; etc. Ajoutons à cela que même les principaux moteurs de recherche ne recensent pas l'intégralité du web et ne peuvent assurer une mise à jour quotidienne de leurs informations (Zaier, 2002).

Une évolution des moteurs de recherche et autres répertoires est donc souhaitable. Des outils se développent peu à peu et ont pour objectif d'automatiser certaines des tâches remplies, actuellement, par les internautes en quête d'informations sur le réseau. Ces nouveaux outils tendent vers un idéal qui est celui d'un outil « intelligent » qui assisterait l'internaute dans ses recherches, les rendant à la fois plus rapides et plus efficaces. Parmi les solutions proposées sur le marché, nous pouvons distinguer les agents sociables, les agents semi-intelligents, les agents de navigation en local (*offline agents*), les agents guides, la webdiffusion (*push media* ou *webcasting*) (Zaier, 2002). Néanmoins, pendant les dix dernières années, les systèmes de recommandation se sont imposés comme moyen efficace pour réduire la complexité dans la recherche d'informations (Konstan, 2004). Ces systèmes apportent la promesse de recherches plus personnalisées et plus optimales. En effet, les systèmes de recommandation, en opposition aux moteurs de recherche et autres répertoires, essayent de proposer à l'utilisateur l'information qui correspond le plus à son profil. Ce profil peut être basé sur les modèles de comportement, l'interaction de l'utilisateur, ou l'information démographique (Zaier, Godin et Faucher, 2008b).

1.2. Problèmes de recherche

Les systèmes de recommandation sont nés de la volonté de pallier le problème de surcharge d'information du web et de certaines contraintes auxquelles se heurtent les outils existants de recherche d'informations. Combinant des techniques de filtrage d'information, personnalisation, intelligence artificielle, réseaux sociaux et interaction personne-machine, les systèmes de recommandation fournissent à des utilisateurs des suggestions qui répondent à leurs besoins et préférences informationnels. En effet, les systèmes de recommandation

sont particulièrement sollicités dans les applications de commerce électronique. Par exemple, *Amazon* recommande toutes sortes de produits (films, musiques, livres, etc.) ; *Netflix*, *Moviecritics*, et *Cinemax* suggèrent des films ; *TiVo* recommande des émissions de télévision ; *iTunes*, *Last.fm*, *Pandora* et *Rhapsody* proposent de la musique ; *Trabble* conseille des restaurants ; *Expedia* et *Travelocity* recommandent des voyages ; *CNN* suggère des nouvelles ; *Jester* propose des blagues ; *Google* recommande des pages web ; *WikiLens* (www.wikilens.org) est une plateforme wiki expérimentale qui permet aux utilisateurs de recommander tout ce qu'ils désirent (Frankowski *et al.*, 2007 ; Zaier, Godin et Faucher, 2008a).

Malgré leur popularité croissante, les systèmes de recommandation ont subi quelques ratés. L'exemple le plus révélateur est l'article anecdotique du *Wall Street Journal* intitulé « *If TiVo Thinks You Are Gay, Here's How to Set It Straight* ». Cet article décrit la frustration des utilisateurs résultant des mauvaises suggestions fournies par le système de recommandation d'émissions de télévision *TiVo* (Tintarev et Masthoff, 2007). Une revue de la littérature nous a permis d'identifier plusieurs problèmes dont souffrent les systèmes de recommandation :

Masse critique. Cet aspect illustre la difficulté à gérer le fait qu'il existe peu d'articles effectivement évalués, ou peu d'utilisateurs qui procèdent à ces évaluations. De ce fait, le système de recommandation ne possède que peu ou pas de données pour générer ces prédictions. Ainsi, il faut dépasser un nombre suffisant d'évaluations avant lequel les recommandations, fournies par le système, ne sont pas pertinentes (Adomavicius et Tuzhilin, 2005 ; Huang, Chen et Zeng, 2004 ; Vellino et Zeber, 2008 ; Ziegler, 2005).

Démarrage à froid. Souvent, on se retrouve confronté au problème qu'un utilisateur ne soit comparable avec aucun autre. Ce problème est dû au fait que peu ou pas d'utilisateurs ont évalué un article donné, ou qu'un utilisateur donné a évalué très peu ou pas d'articles. Généralement, ce problème survient quand un nouvel utilisateur ou une nouvelle ressource est ajouté à la base de recommandation (Adomavicius et Tuzhilin, 2005 ; Condliff, Lewis et Madigan, 1999 ; Leung, Chan et Chung, 2007 ; Schein *et al.*, 2002).

Principe d'induction. Les systèmes de recommandation se basent sur le principe qu'un utilisateur qui a exhibé un comportement dans le passé tendra à exhiber un comportement semblable dans le futur (Vellino et Zeber, 2008). Cependant, ce principe n'est pas nécessairement valable dans le contexte réel. En effet, un utilisateur peut changer complètement de domaine d'intérêt ou en avoir plusieurs. Pour faire face à ce problème, des techniques de dérive d'intérêt (*Interest drift*) ou de changement de contexte (*Context shifts*) ont vu le jour (Bell, Koren et Volinsky, 2007c ; Vellino et Zeber, 2008).

Architecture centralisée vs Architecture distribuée. Les systèmes de recommandation, tels qu'*Amazon*, sont typiquement basés sur une architecture centralisée. Comme la figure 1.1 le montre, le moteur de recommandation centralisé permet de sauvegarder les profils d'utilisateurs et le calcul des recommandations dans un serveur central. Cependant, en dépit de leur popularité, les systèmes de recommandation centralisés souffrent de plusieurs problèmes : le coût, la robustesse, la sécurité, la mobilité, etc. Une des solutions à ces problèmes est de répartir le système. Un système de recommandation distribué pourrait être conçu pour tirer profit de la puissance de calcul disponible sur les ordinateurs des utilisateurs (Miller, Konstan et Riedl, 2004 ; Miller, 2003 ; Ziegler, 2005).

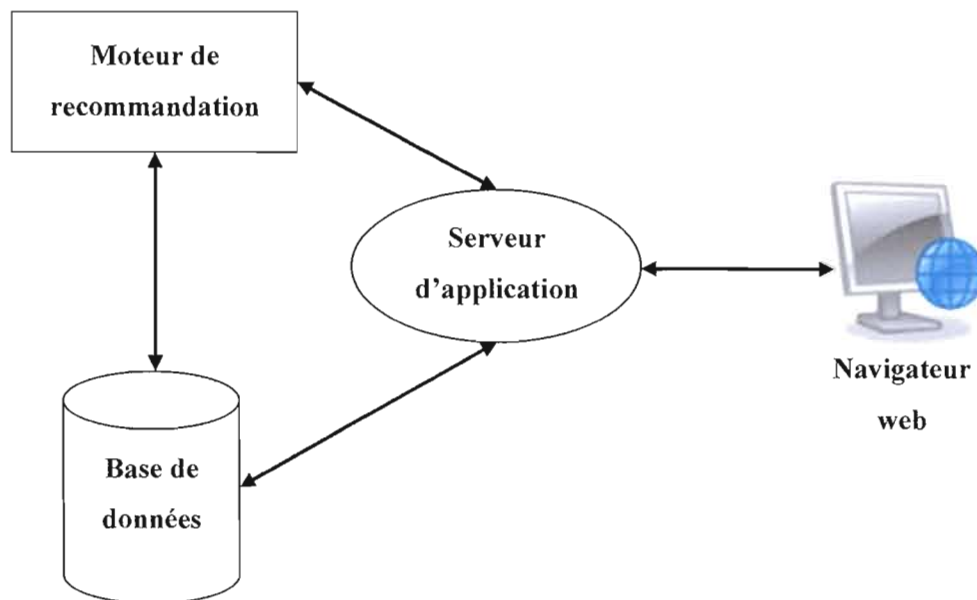


Figure 1.1 Système de recommandation centralisé

Mobilité. L'exemple le plus représentatif du concept de mobilité est le système Magellan de la chaîne d'épicerie Safeway Inc. Le système Magellan, représenté dans la figure 1.2, consiste en un chariot d'épicerie muni d'un ordinateur, un écran à cristaux liquides et une connexion Bluetooth. Ce système propose des produits aux utilisateurs en se basant sur leur proximité à un produit et leur historique d'achats. Cependant, ce prototype présente deux problèmes. D'abord, le système Magellan est basé sur une architecture centralisée et donc souffre des mêmes problèmes évoqués précédemment. En second lieu, le système est limité aux achats en épicerie. L'idée derrière la mobilité est de pouvoir avoir accès à des recommandations à tout moment et en tout endroit (épicerie, film, musique, électronique, etc.) en utilisant son cellulaire, PDA, PALM, etc. (Miller, Konstan et Riedl, 2004 ; Miller, 2003 ; Ziegler, 2005).



Figure 1.2 Magellan

Sécurité et crédibilité. Les systèmes de recommandation ne peuvent pas empêcher les actes de tromperie. Il devient ainsi facile de se forger une nouvelle identité et de se livrer au vandalisme, comme fournir de mauvaises informations au système. Il est à noter que dans les

systèmes distribués exempts d'autorités centrales, il est plus difficile de contrôler l'identité des utilisateurs et de pénaliser le comportement malveillant. Une étude a montré qu'un pourcent de profils erronés suffit à pénaliser grandement un système de recommandation (Burke *et al.*, 2006). Par conséquent, il est indispensable d'avoir des moyens permettant à chaque utilisateur de décider en quels utilisateurs et en quels contenus avoir confiance (Massa et Avesani, 2004, 2007a, 2007b ; Memmi et Nérot, 2003).

Explication de la recommandation. Il faut présenter la recommandation pour qu'elle soit acceptée par l'utilisateur et pour améliorer les recommandations futures (Tintarev et Masthoff, 2007). Dans ce but, il est possible d'utiliser, comme *Amazon* (illustré par la figure 1.3), une explication sous forme de texte ou de graphiques. (Vellino et Zeber, 2008).



Figure 1.3 Explication de la recommandation d'*Amazon* (Vellino et Zeber, 2008)

Collecte des préférences. Une des étapes les plus importantes et les plus difficiles des systèmes de recommandation est la collecte des préférences des utilisateurs. En effet, l'obtention des évaluations de la part des utilisateurs sur une ressource donnée qui leur a plu, moins plu, ou pas du tout plus, est une tâche ardue. Ainsi, des techniques de collecte des préférences utilisateur, intrusives ou pas, ont vu le jour (Chan, 1999 ; Claypool *et al.*, 2001 ; Miller, 2003).

Complexité computationnelle. L'évaluation des similarités, une partie intégrale du filtrage de collaboration, implique quelques processus comportant de nombreux calculs. Pour un nombre important d'utilisateurs, le calcul de la similarité pour tous les individus du voisinage devient infaisable. Ainsi, une bonne performance peut seulement être assurée en limitant ces calculs. Pour ce faire, il est important de restreindre suffisamment la taille de la communauté. Ainsi, des mécanismes intelligents de filtrage sont nécessaires ne sacrifiant pas trop d'information importante (Zaier, Godin et Faucher, 2008b ; Ziegler, 2005).

Protection de la vie privée. Un autre problème qui touche les systèmes de recommandation est la protection des informations sensibles constituant le profil utilisateur (information personnelle, intérêts, goûts, habitudes, etc.). Vu la nature de l'information, ces systèmes doivent assurer une telle protection. Ainsi, des moyens, pour préserver l'anonymat des utilisateurs et chiffrer les données transmises, sont nécessaires (Aimeur *et al.*, 2006 ; Aimeur *et al.*, 2008 ; Kobsa, 2007 ; Lam, Frankowski et Riedl, 2006 ; Memmi et Nérot, 2003).

1.3. Proposition de recherche

Dans cette section, nous commencerons par décrire l'approche proposée pour pallier les problèmes mentionnés ci-dessus. Ensuite, nous discuterons des contributions apportées par notre recherche.

1.3.1. Approche proposée

Dans cette thèse, nous nous intéressons à certains des problèmes évoqués dans la section précédente, à savoir la masse critique, le principe d'induction, l'architecture du système, la

collecte des préférences et finalement, la complexité computationnelle. À cet effet, nous proposons un agent de filtrage d'information coopératif, basé sur une architecture pair-à-pair, capable de comportements sociaux complexes. Plus précisément, nous allons approfondir les points suivants :

Filtrage de l'information. Nous nous proposons de développer un système capable de prédire dans un ensemble de ressources disponibles celles qui sont susceptibles d'intéresser un utilisateur mettant à profit l'information sur la ressource, les modèles de comportement, l'interaction de l'utilisateur, et l'information démographique.

Plate-forme d'agents adaptables. Notre modèle d'agents adaptables propose des mécanismes qui permettent aux agents de réagir, plus facilement et plus rapidement, aux changements qui surviennent dans son environnement. L'objectif central de ce modèle est de permettre aux agents d'adopter des comportements spécifiques en fonction de l'évolution de leur environnement.

Collaboration entre agents autonomes. Notre approche, pour permettre aux agents d'interagir avec leur environnement, consiste dans l'utilisation d'agents autonomes très simples. Ces derniers collaborent pour résoudre des tâches ardues. Le but principal de cette approche est de permettre l'émergence de comportements sociaux complexes au sein de notre système multi-agents.

Adaptation de divers profils utilisateurs basée sur l'apprentissage. L'apprentissage automatique de divers profils utilisateurs s'inscrit dans le cadre de cette recherche. Le but de ce concept est de trouver un moyen pour mieux caractériser chaque utilisateur. Ce dernier nous permettra de recommander plus efficacement à l'utilisateur les ressources qui seront susceptibles de l'intéresser.

1.3.2. Contributions

Les contributions apportées par cette thèse se rapportent à diverses disciplines, à savoir, filtrage et repérage de l'information, systèmes distribués, réseaux pair-à-pair, agents intelligents, comportements sociaux, etc. Il est à noter que toutes les contributions

mentionnées ci-dessous ne se limitent pas à notre système de recommandation distribué, mais elles sont également importantes pour de nombreuses autres applications, comme les systèmes multi-agents, les gestionnaires de profils utilisateurs, la réduction de la complexité computationnelle, la collecte des préférences utilisateurs, le PageRank, etc.

État de l'art des systèmes de recommandation. Un des objectifs de cette thèse est de dresser l'état de l'art sur les systèmes de recommandation, de distinguer les forces et faiblesses des techniques et algorithmes, de filtrage de l'information collaboratif, existants, et finalement, d'identifier les bonnes pratiques, connues et éprouvées qui permettent d'atteindre de meilleures recommandations.

Modèle d'évaluation expérimentale. Ce modèle d'évaluation expérimentale repose sur l'emploi de groupes d'utilisateurs de taille variable. Ce dernier représente une simulation de la recherche du voisinage dans une architecture distribuée (Zaier, Godin et Faucher, 2008b). De plus, une récente étude a mis en évidence l'impact de la taille du voisinage sur différents algorithmes de recommandation (Zaier, Godin et Faucher, 2007). En outre, ce modèle expérimental implique l'identification d'un corpus expérimental qui suit une distribution réelle et adaptable à notre algorithme de recommandation hybride (Zaier, Godin et Faucher, 2008a). Enfin, ce modèle présuppose la détermination des métriques d'évaluation adaptables à plateforme de recommandation distribuée.

Profils utilisateurs et calcul de similarité. Notre approche, pour créer les profils utilisateurs, est basée sur l'information de la ressource à évaluer, l'interaction de l'utilisateur, et l'information démographique. Les profils en résultant sont utilisés pour le calcul de la similarité entre utilisateurs. De même, des profils initiaux, fournis par les utilisateurs, permettent d'éviter les problèmes de masse critique et de démarrage à froid. De plus, l'utilisation de profils multiples pour chaque utilisateur nous permettra de pallier le problème du principe d'induction (Zaier, Godin et Faucher, 2007, 2008b).

Comportements sociocognitifs dans les systèmes multi-agents. Cette thèse s'intéresse au problème de l'organisation dans les «systèmes ouverts». Dans ce cadre, cette recherche vise à définir des modèles de comportements sociaux permettant aux agents de s'organiser

pour s'adapter à leur environnement. Pour ce faire, nous procédons, dans un premier temps, à une revue de littérature visant à recenser les différents comportements sociocognitifs chez les animaux. Ensuite, nous allons identifier les différents comportements, parmi les comportements préalablement identifiés, susceptibles d'être produits par notre système multi-agents. Cette complexification des agents enrichit leur comportement et donc les possibilités d'interaction. Cela donne accès à certaines caractéristiques sociales, telles que le regroupement et la spécialisation, le partage des ressources, etc. Notez cependant que pour des raisons de temps, nous n'explorerons pas cet aspect des agents dans notre partie plus empirique. Nous remettons à plus tard la mise au point d'un système qui pourrait tenir compte des comportements sociaux complexes des agents.

Algorithme de filtrage collaboratif. Nous avons intégré une technique de discrimination des pairs dans la formation du voisinage, pour le calcul des recommandations. Des évidences empiriques montrent que le système en résultant permet de réduire la complexité computationnelle dans le calcul de similarité et le bruit dans les systèmes de recommandation (Zaier, Godin et Faucher, 2008b).

Plateforme distribuée de recommandation. Notre dernière contribution vise à intégrer toutes les techniques et discussions des contributions précédentes dans une infrastructure de recommandation distribuée. Des résultats empiriques sont présentés afin de comparer l'efficacité de notre approche à deux algorithmes de recommandation (Zaier, Godin et Faucher, 2008b).

1.3.3. Travaux publiés

Une grande partie des contributions apportées dans cette thèse a été publiée dans des conférences internationales (Zaier, Godin et Faucher, 2007, 2008b, 2008a). Dans un premier article (Zaier, Godin et Faucher, 2007), nous avons pu étudier l'impact de la taille du voisinage sur la performance de différentes méthodes de filtrage collaboratif. Notre deuxième publication (Zaier, Godin et Faucher, 2008b) introduit, dans un premier temps, un algorithme de recommandation basé sur la discrimination du voisinage. Ensuite, nous avons démontré l'incidence de cette approche sur la qualité des recommandations. Finalement,

notre troisième et dernière publication (Zaier, Godin et Faucher, 2008a) procède, tout d'abord, à une revue des différents corpus utilisés dans l'évaluation des systèmes de recommandation. Subséquemment, nous démontrons la corrélation entre la distribution des évaluations dans le corpus et la performance du système de recommandations.

1.4. Présentation de la recherche

L'objectif principal de cette thèse est de développer et d'évaluer un modèle, basé sur une architecture pair-à-pair, pour le filtrage collaboratif de l'information exploitable dans un contexte Internet. À cette fin, le reste de ce document est organisé en deux parties.

La première regroupe les aspects théoriques de notre travail. Nous nous proposons de fournir quelques pistes de recherche pour répondre à notre problématique. En effet, lors de notre revue de la littérature, nous identifions les travaux les plus pertinents pour notre sujet de recherche. Ainsi, dans cette partie, nous passerons en revue brièvement les caractéristiques des modèles de systèmes de recommandation qui existent dans la littérature. Ensuite, nous identifierons les points forts et les faiblesses de chaque modèle. Enfin, nous discuterons des améliorations possibles que nous pourrions y faire.

Étant donné que le sujet choisi est tout à fait interdisciplinaire, les documents étudiés viennent de différents champs de recherche, à savoir : le filtrage l'information, les architectures distribuées, les systèmes multi-agents et les comportements sociaux. Le chapitre 2 présente les caractéristiques générales des systèmes de recommandation, ainsi que leurs forces et faiblesses. Les systèmes de recommandation distribués et les différentes architectures pair-à-pair utilisées sont présentés dans le chapitre 3. Le chapitre 4 s'intéresse aux agents sociaux de recommandation, puis en s'inspirant des travaux en éthologie, aux différents phénomènes sociaux susceptibles d'être produits par ces derniers.

La deuxième partie de cette thèse se concentre sur le modèle qui résulte de la partie théorique. En effet, les conclusions des chapitres précédents nous auront permis d'identifier quelques solutions susceptibles de produire des résultats intéressants. Le chapitre 5 présente l'approche proposée de modèle coopératif, basé sur une architecture pair-à-pair, capable de comportements sociaux complexes. Le chapitre 6 présente le dispositif expérimental utilisé

pour valider le modèle proposé, ainsi que les méthodes de mesure et le corpus de travail. Nous présentons donc les résultats de l'étude expérimentale. Enfin, nous résumons dans le dernier chapitre, l'apport de ce travail de thèse et nous proposons des perspectives d'évolution tant à court qu'à long terme.

Première partie
Étude Théorique

Chapitre II

Filtrage de l'information

« Pour prévoir l'avenir, il faut connaître le passé. »

– Nicolas Machiavel (1469 – 1527)

2.1. Introduction

Généralement, l'accès à l'information sur Internet se fait des deux façons suivantes. Une recherche active de l'information est accomplie par l'intermédiaire d'outils de recherche d'informations. Ces derniers ont pour objectif de fournir à l'utilisateur les documents¹ qui vont satisfaire la requête de recherche formulée (Belkin et Croft, 1992). L'accès à l'information jugée pertinente se fait par des systèmes de filtrage d'information. À l'opposé des outils de recherche d'informations, le filtrage d'information ne requiert pas une formulation systématique du besoin informationnel de l'utilisateur. Ainsi, cette approche permet notamment de réduire la complexité de la recherche d'informations et facilite l'accès à l'information. Le système de filtrage d'informations achemine l'information aux utilisateurs en se basant sur leurs profils. Ces derniers sont établis grâce à des techniques d'apprentissage des goûts des utilisateurs (Belkin et Croft, 1992). Nous présentons ci-dessous un tableau comparatif détaillé des différences entre la recherche d'informations et le filtrage d'informations.

¹ Les recommandations fournies par un système de filtrage d'information ne se limitent pas à des documents. Néanmoins, pour une meilleure compréhension, nous prenons l'exemple d'un système de recommandation de documents.

Tableau 2.1 Recherche d'informations versus filtrage d'informations (Berrut et Denos, 2003)

	Recherche d'informations	Filtrage d'information
Approche	Trouver l'information recherchée	Filtrer l'information non désirée
Livraison	Corpus statique, sur demande	Flux dynamique
Persistance	Des besoins à court terme	Des intérêts à long terme
Analyse du contenu	Utilise des mots-clés	Pour le filtrage basé sur le contenu. Pas pour le filtrage collaboratif.
Fonctionnalités	Non personnalisé, non adaptatif, non dynamique, à court terme.	Personnalisé, s'attaquent aux changements du profil de l'utilisateur, filtre dynamiquement l'information entrante, à long terme.

La figure ci-dessous nous présente un modèle général de filtrage d'information. Comme nous l'avons vu plus haut, les systèmes de filtrage impliquent l'existence d'un flux d'information entrant qui est produit par une source d'information distante ou envoyé directement par d'autres sources (d'autres utilisateurs). En ce qui concerne le filtrage, il est basé sur des caractérisations d'utilisateurs ou de groupes d'utilisateurs constituant des profils. Ces derniers représentent, généralement, un ensemble de domaines d'intérêt. Enfin, l'information résultante du filtrage est évaluée par les utilisateurs, pour mettre à jour leurs profils et leurs domaines d'intérêt.

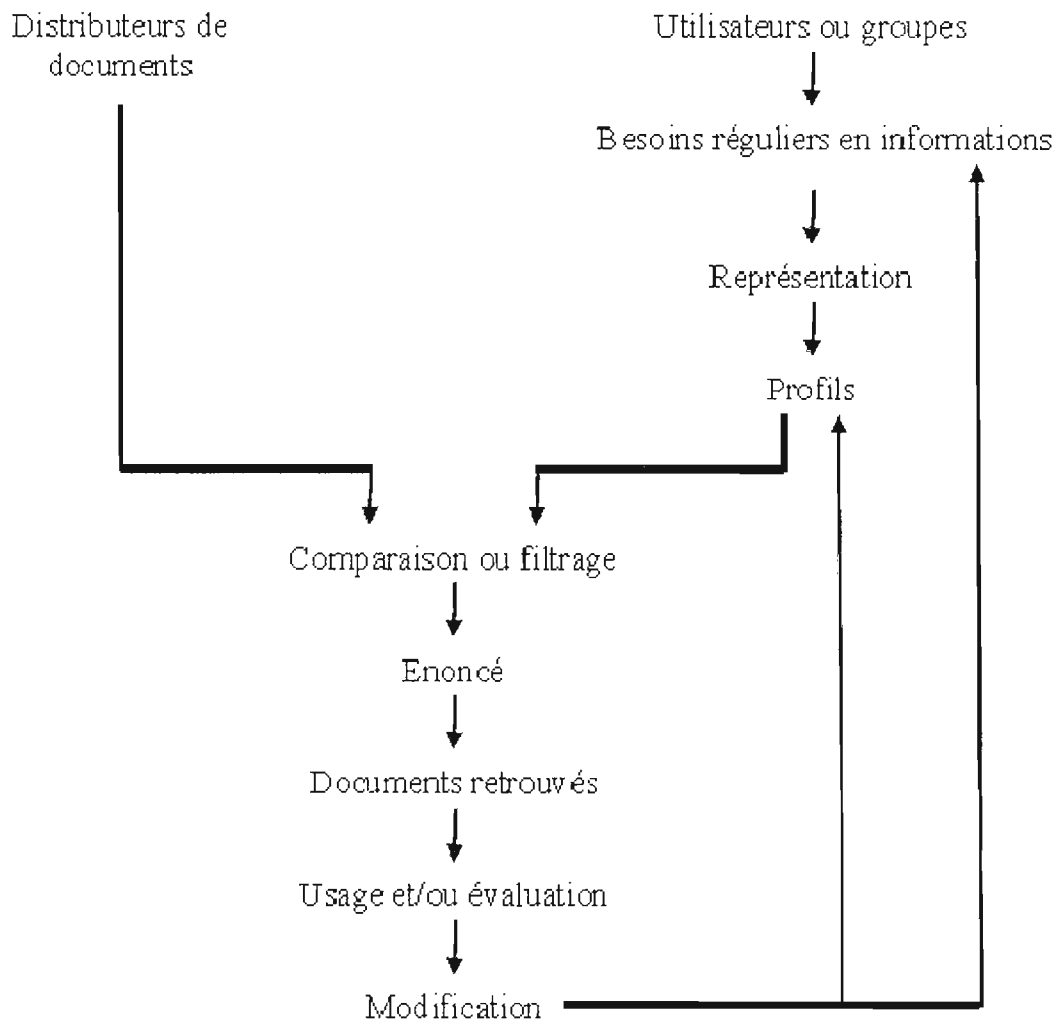


Figure 2.1 Modèle général pour le filtrage d'information, inspiré de Berrut et Denos (2003)

2.2. Collecte des préférences

Comme nous l'avons évoqué dans la section précédente, les systèmes de filtrage présupposent l'existence d'un système de collecte du degré d'appréciation des utilisateurs pour une ressource donnée. En effet, pour caractériser un profil d'utilisateur, la première chose à collecter est l'évaluation d'une ressource donnée faite par un utilisateur : si cette dernière lui a plu, moins plu, ou pas du tout plus. À cet effet, il existe deux approches :

demander à l'utilisateur explicitement d'évaluer lui-même les ressources à partir d'une échelle de notes préalablement définie, ou bien inférer automatiquement cette évaluation, grâce aux informations que l'on peut récolter à partir des données du protocole de communication (Claypool *et al.*, 2001).

Évaluations explicites. Les utilisateurs sont obligés de spécifier explicitement leurs préférences pour toutes les ressources. Généralement, les utilisateurs doivent indiquer leur degré d'appréciation sur une échelle de 1 à 5 ou 1 à 7. De récentes études ont mis en évidence l'impact du choix de l'échelle d'évaluation sur la qualité des recommandations (Ziegler, 2005).

Évaluations implicites. Les évaluations explicites imposent aux utilisateurs des efforts supplémentaires. En conséquence, les utilisateurs tendent souvent à éviter ce fardeau en quittant définitivement le système ou en fournissant des évaluations erronées (Ziegler, 2005). À l'opposé, la déduction de ces évaluations par la seule observation du comportement des utilisateurs est beaucoup moins intrusive. Parmi les exemples typiques de ces observations pour des évaluations implicites, on peut citer l'historique d'achat pour *Amazon*, le temps de lecture pour *Usenet News* (Ziegler, 2005), et l'analyse des comportements de navigation sur Internet pour *Quickstep* (Gaul et Schmidt-Thieme, 2002 ; Middleton, Shadbolt et De Roure, 2004).

Un exemple réel de l'inférence des évaluations implicite est la formule, proposée par Chan (1999), pour prédire si une page web a été appréciée ou pas. Cette formule se base, en grande partie, sur les informations que l'on peut récolter à partir des données du protocole de communication. En effet, elle est calculée en fonction de l'historique, le marque-page (*bookmark*), le contenu des pages et le journal des accès (« *access log* »).

Plus précisément, l'historique d'un navigateur web maintient une trace du dernier moment qu'une page a été visitée. On peut donc utiliser cette donnée pour calculer combien de fois un utilisateur passe sur une page et depuis quand il n'est pas retourné la visiter. De plus, on peut également supposer qu'une URL présente dans le marque-page d'un utilisateur est considérée comme intéressante pour celui-ci. En outre, chaque page contient des liens vers d'autres pages. On suppose que si un utilisateur est intéressé par une page, il va

probablement visiter les liens référencés par celle-ci. Ainsi, on conjecture qu'un fort pourcentage de liens visités à partir d'une page dénote un intérêt particulier pour cette page. Enfin, chaque entrée dans un journal des accès correspond à une requête HTTP, qui contient typiquement l'adresse IP du client, une marque du moment de connexion, des méthodes d'accès, une URL, un protocole, un statut, et une taille de fichier. Typiquement, une ligne de fichier de log se présente comme suit:

« 216.22.34.11 - - [10/Jul/2008:04:04:54 +0200] "GET /uqam.ca/index.html HTTP/1.1"
200 2856 »

Où

- ✓ 216.22.34.11 correspond à l'adresse IP du client,
- ✓ 10/Jul/2008:04:04:54 correspond au moment de connexion,
- ✓ GET correspond à la méthode d'accès,
- ✓ /uqam.ca/index.html est l'URL de la page qui a été visitée,
- ✓ HTTP/1.1 correspond au protocole de communication,
- ✓ 200 est la valeur de retour (ici, tout s'est bien passé),
- ✓ 2856 correspond à la quantité d'informations transférées (généralement la taille du fichier).

Grâce à ces informations, le temps passé sur chaque page peut être calculé. Cependant, le temps passé sur une page dépendant également de la longueur de cette page, l'intérêt d'un utilisateur pour une page sera calculé par le temps passé sur la page, normalisé par la taille de la page.

Finalement, Chan (1999) définit le degré d'intérêt d'une page par:

$$\begin{aligned}
 \text{Interest}(\text{Page}) &= \text{Frequency}(\text{Page}) \\
 &\quad * (1 + \text{IsBookmark}(\text{Page}) + \text{Duration}(\text{Page}) + \text{Recency}(\text{Page}) \\
 &\quad + \text{LinkVisitPercent}(\text{Page}))
 \end{aligned}$$

Où

- ✓ $Frequency(Page)$ est la fréquence de visite de la page;
- ✓ $IsBookmark(Page) = \begin{cases} 1 & \text{si la page appartient au marque - pages} \\ 0 & \text{sinon} \end{cases}$
- ✓ $Duration(Page) = \frac{TotalDuration(Page) / Size(Page)}{\max_{Page \in VisitedPage} TotalDuration(Page) / Size(Page)}$
- ✓ $Recency(Page) = \frac{Time(LastVisit) - Time(StartLog)}{Time(Now) - Time(StartLog)}$
- ✓ $LinkVisitPercent(Page) = \frac{NumberOfLinksVisited(Page)}{NumberOfLinks(Page)}$

Un autre exemple concret pour l'évaluation non intrusive est la collecte des préférences des utilisateurs sur la musique écoutée. Les lecteurs multimédias, comme *iTunes* et *Windows Media Player*, ont la possibilité d'observer le nombre de fois que vous écoutez une chanson. Ces logiciels peuvent aussi se rappeler la dernière fois que vous avez écouté chaque chanson. Ces informations peuvent être utilisées pour dégager les évaluations implicites des utilisateurs (Miller, 2003). La figure 2.2 illustre les différents aspects de notre vie quotidienne qui peuvent être utilisés par les systèmes de recommandations.

Pour conclure, malgré la facilité qu'elle peut apporter dans la collecte des préférences des utilisateurs, l'évaluation implicite comporte quelques limites. Par exemple, les achats faits pour offrir à une autre personne (achats de cadeaux) ou pour le compte de quelqu'un d'autre (achats pour son entreprise) ne reflètent pas les intérêts de l'utilisateur. Ainsi, les évaluations implicites ne représentent pas toujours les préférences réelles de l'utilisateur. De plus, il est difficile d'obtenir des évaluations explicites des utilisateurs. Pour ces raisons, un certain nombre de systèmes de recommandation adoptent des approches hybrides. Amazon.com calcule ses recommandations en se basant, autant que possible, sur des évaluations explicites. En cas d'indisponibilité de ces dernières, des évaluations implicites, inférées des observations des comportements de l'utilisateur, sont utilisées à la place (Ziegler, 2005).

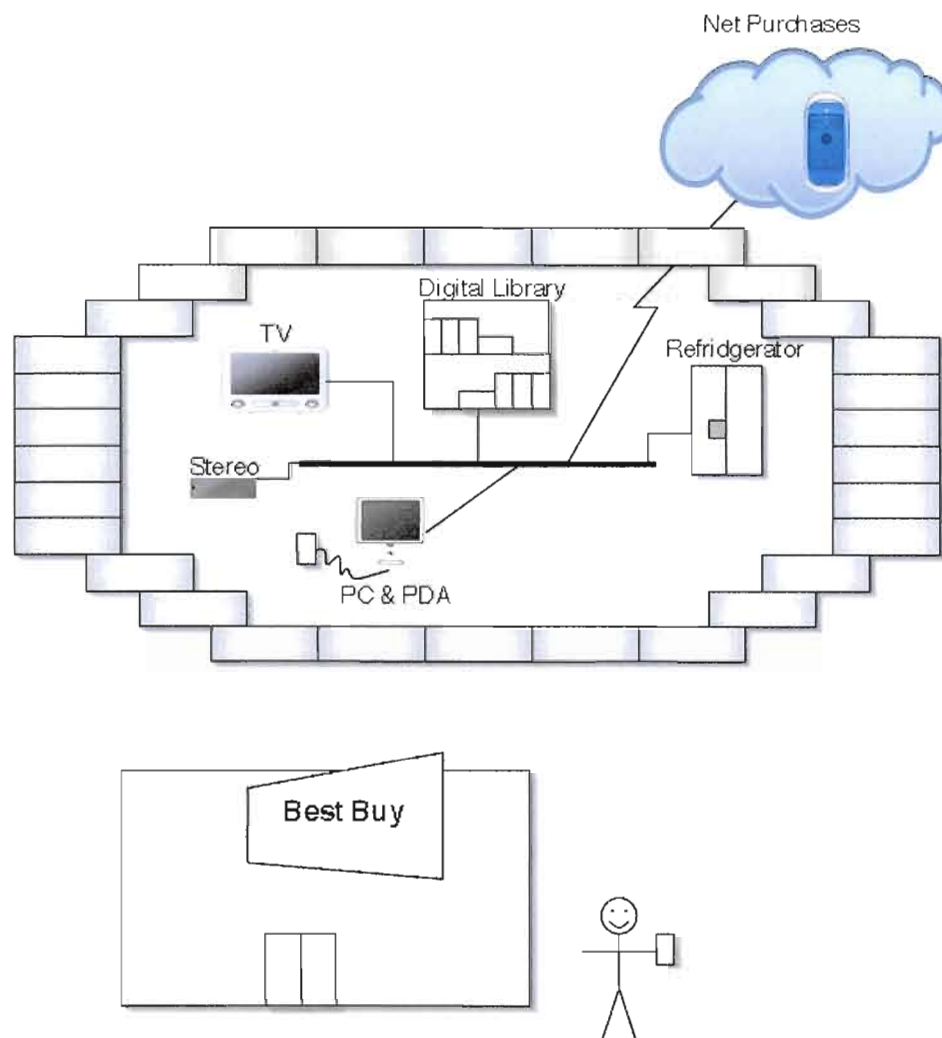


Figure 2.2 Aspects de vie quotidienne pouvant être utilisés par les systèmes de recommandation (Miller, 2003)

2.3. Grandes familles de filtrage

Une revue de la littérature montre qu'il existe trois grandes familles de systèmes de filtrage d'information (Adomavicius et Tuzhilin, 2005):

Le filtrage basé sur le contenu. Dans cette approche aussi appelée filtrage cognitif, le choix des documents à recommander est basé sur une comparaison des thèmes abordés dans les documents par rapport aux thèmes intéressant l'utilisateur.

Le filtrage collaboratif. Pour cette approche également appelée système de recommandation, la sélection des documents est, généralement, basée sur les appréciations des utilisateurs, jugés semblables, des documents.

Le filtrage hybride. La méthode de filtrage hybride combine les deux approches précédentes.

2.3.1. Filtrage basé sur le contenu

L'idée principale sur laquelle cette approche se base est l'utilisation du contenu des documents lus par les utilisateurs pour aider à la caractérisation de ces derniers. En effet, le filtrage basé sur le contenu repose sur un profil thématique décrivant les besoins de l'utilisateur. Ainsi, cette approche fonctionne comme une requête destinée à un outil de recherche d'informations. Le profil utilisateur joue le rôle d'une requête de recherche utilisée pour filtrer les documents (Adomavicius et Tuzhilin, 2005 ; Berrut et Denos, 2003). Pour ce faire, le filtrage basé sur le contenu doit remplir deux fonctions fondamentales :

- ✓ L'identification des documents pertinents pour l'utilisateur en fonction de son profil.
- ✓ La mise à jour du profil utilisateur en fonction des évaluations, implicites ou explicites (section 2.2), de l'utilisateur des documents reçus.

Plus précisément, il faut procéder à une phase de prétraitement pour transformer le document qu'on veut analyser sous une forme compréhensible par l'algorithme de filtrage (Berrut et Denos, 2003). Dans ce but, il faut dans un premier temps choisir le niveau d'analyse linguistique à utiliser. En effet, différents niveaux d'analyse sont disponibles. Nous pouvons décider de ne pas rentrer en profondeur dans le texte, et de ne considérer que certaines informations superficielles, comme le titre ou les mots clés associés au document. De même, nous pouvons considérer le texte intégral (« *bag of words* ») en supposant que tous les mots sont importants et indépendants les uns des autres. De plus, nous pouvons considérer des *n-grammes*. Plus précisément, nous pouvons utiliser des séquences de *n* caractères. Enfin, nous pouvons également faire appel à des informations syntaxiques

(données par des grammaires) ou sémantiques (données par un *thesaurus*), pour cibler des thèmes plus généraux découlant du texte (Adomavicius et Tuzhilin, 2005 ; Zaier, 2002).

Ensuite, il faut normaliser le texte obtenu. Plus précisément, il faut se débarrasser des mots qui ne sont pas porteurs de sens, comme les caractères spéciaux et les mots dits vides (pronoms, prépositions, conjonctions, articles, etc.). Alors, sur les mots qui restent, nous effectuons une troncature. En d'autres termes, il faut effectuer une analyse des formes morphologiques des termes, pour ne garder que leur racine (par exemple toutes les déclinaisons possibles des verbes, des adjectifs et des noms). Enfin, nous regroupons les mots identiques, en les comptant ou pas, selon l'algorithme qu'on veut appliquer. Ainsi, pour stocker en mémoire le document analysé, la représentation la plus utilisée est le modèle vectoriel. Dans ce modèle introduit par Salton et al. (1975), un document est représenté sous forme d'un vecteur de mots. Nous associons à chaque mot composant le vecteur un poids (« *word frequency weighting* »). Selon l'algorithme choisi, ce dernier représente l'absence ou la présence d'un mot dans un document en fonction de la pertinence de ce dernier (algorithme de *Bayes*), ou le nombre d'occurrences, ou le fait qu'il soit mis en valeur ou pas, *Cosinus*, *Okapi*, *TFIDF* (« *Term Frequency/Inverse Document Frequency* »), etc. (Adomavicius et Tuzhilin, 2005 ; Amati et Joost Van Rijsbergen, 2002 ; Manning, Raghavan et Schtze, 2008 ; Salton, 1989 ; Zaier, 2002).

Une fois cette phase de prétraitement terminée, nous pouvons envisager d'intégrer les mots, qui composent ces documents au profil d'utilisateur. La représentation la plus simple d'un profil d'utilisateur serait alors un vecteur de poids associés à chacun des mots. Alors, nous calculons ces poids, qui caractérisent le profil d'utilisateur par le biais de l'algorithme de *Rocchio* (1971), en fonction des évaluations obtenues en utilisant une des méthodes présentées dans la section 2.2 (Adomavicius et Tuzhilin, 2005 ; Zaier, Godin et Faucher, 2007, 2008b):

$$w_j = \alpha * w_{1,j} + \beta * \frac{\sum_{i \in C} x_{i,j}}{n_c} - \gamma * \frac{\sum_{i \notin C} x_{i,j}}{n - n_c}$$

Où, pour un utilisateur donné j , C est l'ensemble des documents pertinents, n est le nombre total d'articles, n_c est le nombre de documents pertinents, $w_{i,j}$ est le poids existant, le $x_{i,j}$ est le *TF-IDF* de chaque terme du document i , et α , β et γ des constantes. Cependant, plusieurs autres techniques peuvent être utilisées, comme les arbres de décision, la Classification bayésienne, le regroupement (« *clustering* »), les réseaux de neurones (Balabanovi et Shoham, 1997 ; Billsus et Pazzani, 1999, 2000 ; Lang, 1995 ; Mooney, Bennett et Roy, 1998 ; Mooney et Roy, 2000 ; Pazzani et Billsus, 1997 ; Widyanoro, Loerger et Yen, 2001 ; Zhang, Callan et Minka, 2002).

Pour conclure, le filtrage basé sur le contenu présente un certain nombre de limites. Tout d'abord, nous pouvons souligner la difficulté à indexer les documents multimédias. En effet, le profil utilisateur peut prendre diverses formes, mais il demeure toujours composé par des mots. Ces derniers seront comparés aux mots qui composent le document. De ce fait, il est impossible d'indexer des documents multimédias. En outre, le fait que cette approche se base uniquement sur les profils thématiques pour caractériser les utilisateurs, pose également un problème. En effet, le filtrage basé sur le contenu n'intègre pas d'autres critères de pertinence que le profil thématique. Pourtant, il existe de nombreux autres aspects comme les informations démographiques (âge, position géographique, emploi, etc.), la qualité, la confiance, etc. De plus, l'effet dit d'« entonnoir » restreint le champ de vision des utilisateurs. En effet, le profil évolue toujours dans le sens d'une expression du besoin de plus en plus spécifique, qui ne laisse pas de place à des documents pourtant proches, mais dont la description thématique diffère. Par exemple, lorsqu'un nouvel axe de recherche apparaît dans un domaine, avec de nouveaux termes, ces derniers n'apparaissent pas dans le profil. Ainsi, les documents, portant sur cet axe de recherche, seront éliminés automatiquement par le système de filtrage. De ce fait, l'utilisateur n'aura donc jamais l'occasion d'exprimer un retour de pertinence positif sur ce nouvel axe de recherche. Enfin, la masse critique est une autre limite du filtrage basé sur le contenu. En effet, il faut un certain nombre de documents pour que le système commence à performer (Adomavicius et Tuzhilin, 2005 ; Berrut et Denos, 2003).

2.3.2. Filtrage collaboratif

L'hypothèse principale sur laquelle repose le filtrage collaboratif est le bouche à oreille. Plus précisément, les gens à la recherche d'information devraient pouvoir bénéficier de ce que d'autres utilisateurs ont déjà trouvé et évalué. Par exemple, les personnes, qui veulent regarder un film ou lire un livre, demandent à leurs amis leurs opinions. Donc, dans le filtrage collaboratif, la sélection des documents à proposer à un utilisateur ne dépend plus des termes constituant le document (filtrage basé sur le contenu), mais des évaluations faites par les membres de son voisinage. Ainsi, si deux utilisateurs Alice et Bob ont évalué un certain nombre de documents de façon similaire, il y a de fortes chances qu'Alice aime ce que Bob aime, et inversement. Donc les documents que Alice a aimés peuvent être recommandés à Bob et inversement. De la sorte, cette approche résout une difficulté importante rencontrée par l'approche de filtrage basé sur le contenu, à savoir le traitement de documents multimédias (Adomavicius et Tuzhilin, 2005 ; Sarwar *et al.*, 2000a).

La figure 2.3 illustre l'algorithme général d'un système de filtrage collaboratif. Typiquement, les étapes de cet algorithme sont les suivantes (Adomavicius et Tuzhilin, 2005 ; Berrut et Denos, 2003):

- ✓ collecter les appréciations de l'utilisateur sur les documents qu'il consulte;
- ✓ intégrer ces informations dans le profil de l'utilisateur;
- ✓ utiliser ce profil pour aider l'utilisateur dans ces prochaines recherches d'information.

Donc, la première étape à prendre en compte est celle de la collecte du taux de satisfaction des utilisateurs pour les documents qu'ils consultent. Ces évaluations sont recueillies en utilisant une des méthodes présentées dans la section 2.2. Les notes ainsi obtenues sont représentées sous forme d'une matrice. À chaque ligne est associé un utilisateur, et à chaque colonne, un document (Adomavicius et Tuzhilin, 2005). Le tableau 2.2 illustre une matrice d'évaluation pour une base de 5 utilisateurs et 5 documents. Cette matrice correspond formellement à un ensemble d'évaluations v_{ij} faites par des utilisateurs i

sur des documents j . Par exemple, l'évaluation faite par l'utilisateur 3 sur le document 5 est de $7 \rightarrow v_{3,5}=7$. v_i correspond au vecteur décrivant l'utilisateur i . En d'autres termes, v_i correspond à l'ensemble de ses évaluations. Dans notre exemple, nous avons donc $v_3=[0,0,6,6,7]$. On utilise la notation I_i pour désigner l'ensemble des documents que l'utilisateur i a évalué. Nous avons alors $I_3= [3, 4, 5]$. Finalement, nous notons $p_{i,j}$ la prédiction de l'évaluation de l'utilisateur i sur le document j , le but étant que $p_{i,j}$ soit le plus proche possible de $v_{i,j}$.

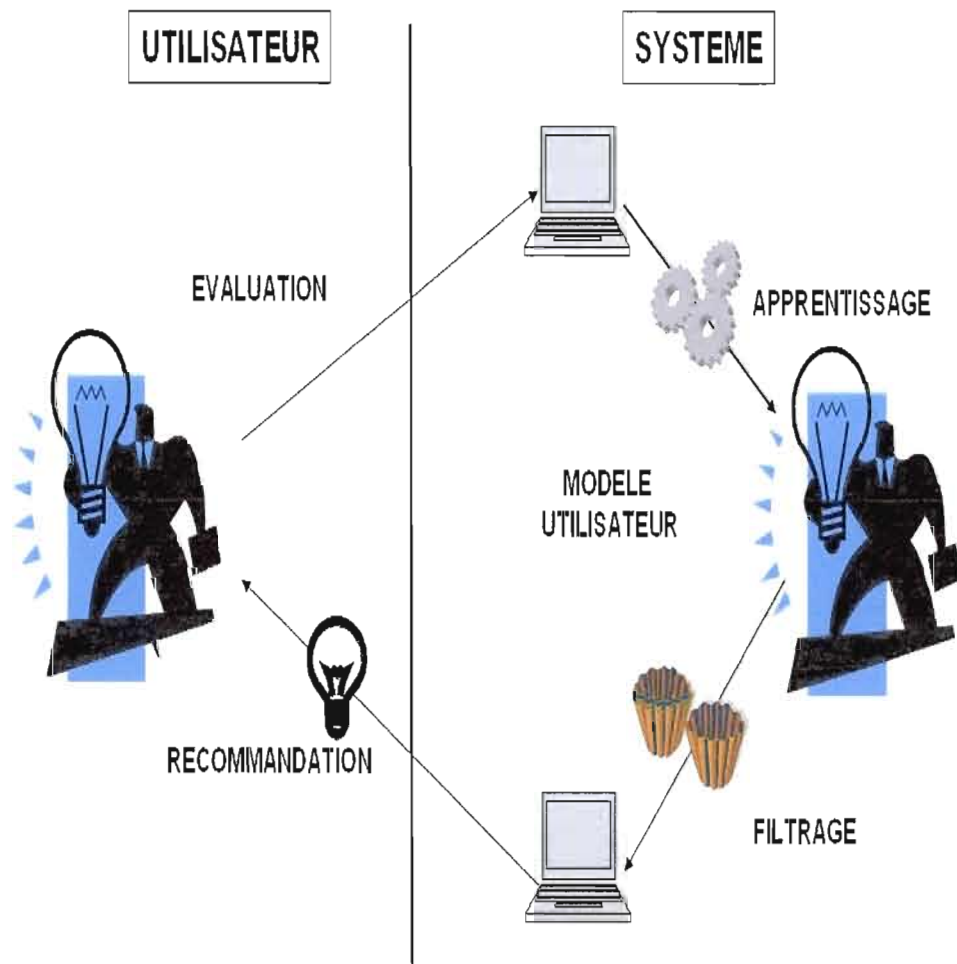


Figure 2.3 Modèle général pour le filtrage collaboratif de l'information (Berrut et Denos, 2003)

Tableau 2.2 Une matrice des notes attribuées aux articles par les utilisateurs

	Document 1	Document 2	Document 3	Document 4	Document 5
Utilisateur 1			7	6	
Utilisateur 2			5	6	7
Utilisateur 3			6	6	7
Utilisateur 4	7	5			7
Utilisateur 5	7	6	?		7

Dans notre exemple, nous voulons prédire l'évaluation de l'utilisateur 5 sur le document 3 $\rightarrow p_{5,3}$. Selon Adomavicius et Tuzhilin (2005), il existe deux grandes familles d'algorithmes de filtrage collaboratif pour calculer cette prédiction : d'une part, les algorithmes basés sur la mémoire (« *Memory-based* » ou « *Heuristic-based* ») utilisent l'intégralité de la matrice d'évaluation pour faire ces prédictions. Dans la phase de prédiction, les utilisateurs ou les articles similaires sont identifiés en se basant sur les évaluations de la matrice. Ensuite, en se basant sur les évaluations de ces utilisateurs ou ces articles similaires, une recommandation pour l'utilisateur peut être générée. À cet effet, plusieurs techniques peuvent être utilisées, comme les plus proches voisins, la théorie des graphes, le regroupement (« *clustering* »), etc. (Aggarwal *et al.*, 1999 ; Breese, Heckerman et Kadie, 1998 ; Delgado, 1999 ; Deshpande et Karypis, 2004 ; Herlocker *et al.*, 1999 ; Hill *et al.*, 1995 ; Jin, Chai et Si, 2004 ; Nakamura et Abe, 1998 ; Pennock *et al.*, 2000 ; Resnick *et al.*, 1994 ; Sarwar *et al.*, 2001 ; Shardanand et Maes, 1995 ; Wang, de Vries et Reinders, 2006 ; Xue *et al.*, 2005). Étant donné que l'algorithme basé sur la mémoire utilise l'intégralité de la matrice d'évaluation pour générer ces recommandations, l'inconvénient de ce dernier est le problème de masse critique.

D'autre part, les algorithmes basés sur les modèles (« *Model-based* ») utilisent la matrice d'évaluation pour générer un modèle, qui sera ensuite utilisé pour les prédictions. Pour ce faire, plusieurs techniques peuvent être utilisées comme les arbres de décision, les modèles probabilistes, les réseaux Bayésiens, les réseaux de neurones, la régression linéaire, le regroupement (« *clustering* »), concept formel, etc. (Adda *et al.*, 2007 ; Billsus et Pazzani, 1998 ; Breese, Heckerman et Kadie, 1998 ; Canny, 2002a ; Chien et George, 1999 ; Getoor et Sahami, 1999 ; Goldberg *et al.*, 2001 ; Hofmann, 2003, 2004 ; Huang, Chen et Zeng, 2004 ; Kumar *et al.*, 1998 ; Marlin, 2003 ; Pavlov et Pennock, 2002 ; Pennock *et al.*, 2000 ; Shani, Brafman et Heckerman, 2002 ; Si et Jin, 2003 ; Ungar et Foster, 1998 ; Wang, de Vries et Reinders, 2008 ; Yu *et al.*, 2004 ; Yu *et al.*, 2002). L'avantage des algorithmes basés sur les modèles est que la complexité computationnelle de ces algorithmes est largement inférieure à celle des algorithmes basés sur la mémoire.

Les approches de filtrage collaboratif sont généralement classifiées en tant qu'algorithmes basés sur les modèles ou algorithmes basés sur la mémoire. Cependant, Wang *et al.* (2006) proposent un autre type de classification de ces algorithmes. Dans cette classification, il y a deux types d'algorithmes. D'une part, les algorithmes basés sur les utilisateurs (« *user-based* ») prédisent l'évaluation d'un utilisateur donné sur un article donné en se basant sur l'information des évaluations pour des profils d'utilisateur similaires (Breese, Heckerman et Kadie, 1998 ; Frankowski *et al.*, 2007 ; Herlocker *et al.*, 1999 ; Jin, Chai et Si, 2004 ; Resnick *et al.*, 1994). D'autre part, les algorithmes basés sur les articles (« *item-based* ») fonctionnent selon le même principe, mais utilisent la similarité entre articles au lieu de la similarité entre utilisateurs (Deshpande et Karypis, 2004 ; Linden, Smith et York, 2003 ; Sarwar *et al.*, 2001).

Pour conclure, le filtrage collaboratif présente un certain nombre de limites. En premier lieu, nous pouvons évoquer le problème lié à la taille de la population d'utilisateurs et de l'ensemble de documents à évaluer (le problème de la masse critique). En effet, il faut un certain nombre d'évaluations avant que le système puisse donner des résultats. De plus, cette approche souffre aussi du problème de démarrage à froid. En effet, les nouveaux utilisateurs commencent par un profil vide. Ainsi, une période d'apprentissage est nécessaire avant que le profil ne reflète concrètement les préférences de l'utilisateur. Pendant cette période, le

système ne peut pas filtrer efficacement. Enfin, comme pour le filtrage basé sur le contenu, cette approche n'intègre pas d'autres aspects de pertinence capables d'améliorer le filtrage (Adomavicius et Tuzhilin, 2005 ; Berrut et Denos, 2003 ; Schein *et al.*, 2002).

2.3.3. Filtrage hybride

L'approche de filtrage hybride repose sur l'idée de tirer profit des avantages des deux approches précédentes, en résolvant les problèmes qui leur sont liés. En fait, ces deux approches paraissent complémentaires. Les chercheurs du domaine estiment que le fait de combiner les deux méthodes pourrait être très bénéfique. D'où l'émergence de plusieurs techniques d'hybridation dont l'objectif consiste à combiner les deux approches (filtrage collaboratif et filtrage basé sur le contenu) de manière efficace (Adomavicius et Tuzhilin, 2005 ; Burke, 2002).

Les récents travaux dans ce domaine visent à développer des algorithmes hybrides de plus en plus efficaces. Burke (2002) décrit sept différents types de méthodes d'hybridation, présentées dans le tableau 2.3. Cependant, selon Adomavicius et Tuzhilin (2005), toutes ces méthodes se basent sur 2 approches principales. La première approche repose sur des méthodes basées sur la mémoire (« *Memory-based* » ou « *Heuristic-based* ») pour générer les recommandations (Balabanovi et Shoham, 1997 ; Basilico et Hofmann, 2004 ; Billsus et Pazzani, 2000 ; Claypool *et al.*, 1999 ; Good *et al.*, 1999 ; Liu et Shih, 2005 ; Melville, Mooney et Nagarajan, 2002 ; Pazzani, 1999 ; Tran et Cohen, 2000 ; Zaier, Godin et Faucher, 2008b). La seconde approche utilise des méthodes basées sur les modèles (« *Model-based* ») pour déterminer les recommandations (Ansari, Essegai et Kohli, 2000 ; Basu, Hirsh et Cohen, 1998 ; Condliff, Lewis et Madigan, 1999 ; Huang, Zeng et Chen, 2004 ; Li et Kim, 2003 ; Popescul *et al.*, 2001 ; Rojsattarat et Soonthornphisaj, 2003 ; Schein *et al.*, 2002 ; Soboroff et Nicholas, 1999 ; Van Setten *et al.*, 2004 ; Vellino et Zeber, 2007 ; Yu, Schwaighofer et Tresp, 2003).

Tableau 2.3 Méthodes d'hybridation (Burke, 2002)

Méthode d'hybridation	Description
Pondérée	Les résultats pondérés de plusieurs techniques de recommandation sont combinés pour produire une nouvelle recommandation.
Permutation	Le système permute entre les différentes techniques de recommandation selon le résultat de la recommandation.
Mixte	Les recommandations de plusieurs techniques sont présentées en même temps.
Combinaison	Différentes techniques de recommandation sont combinées en un unique algorithme de recommandation.
En cascade	Un système de recommandation raffine les résultats fournis par un autre système.
Augmentation	Le résultat (« <i>output</i> ») d'une technique de recommandation est utilisé comme données en entrée (« <i>input</i> ») pour l'autre technique.
Méta-niveau	Le modèle appris par une technique de recommandation est utilisé comme données en entrée (« <i>input</i> ») pour l'autre technique.

2.4. Conclusion

Les deux approches principales (filtrage basé sur le contenu et filtrage collaboratif) de filtrage d'information donnent des résultats très intéressants. Le filtrage collaboratif se propose de recommander aux utilisateurs certains articles qu'ils n'ont pas encore lus en se basant sur les opinions du groupe d'utilisateurs similaires. En revanche, le filtrage basé sur le contenu évalue si un texte est pertinent ou non pour un utilisateur, en fonction de son profil thématique. Néanmoins, ces deux aspects du profil semblent importants. Ainsi, une approche hybride combinant les goûts thématiques personnels et les opinions des utilisateurs du groupe paraît la meilleure méthode pour fournir des recommandations.

Il est à noter que ces techniques de filtrage ne prennent pas en compte un bon nombre d'informations utiles pour expliquer le jugement qu'effectue l'utilisateur. Ceci requiert la prise en compte de différentes sources d'informations sur l'utilisateur (ses préférences précédemment déclarées, son réseau social, etc.) et leurs corrélations avec le contenu des documents qui lui sont proposés. Ainsi, nous avons assisté, ces dix dernières années, à la naissance de plusieurs nouvelles familles de systèmes de filtrage d'information, comme les systèmes de filtrage d'information basés sur l'utilité, les systèmes de filtrage d'information à base d'ontologies, les systèmes d'annotations collaboratives, les systèmes de filtrage d'informations à base de réseaux sociaux, etc. (Cherniack *et al.*, 2003 ; Groh et Ehmig, 2007 ; Lops, Degemmis et Semeraro, 2007 ; Massa et Avesani, 2004 ; Mehta *et al.*, 2005 ; Middleton, Shadbolt et De Roure, 2004 ; Mobasher, Jin et Zhou, 2004 ; Schickel-Zuber et Faltings, 2006 ; Szomszor *et al.*, 2007 ; Tso-Sutter, Marinho et Schmidt-Thieme, 2008 ; Ziegler et Golbeck, 2007 ; Ziegler, Schmidt-Thieme et Lausen, 2004).

Un autre aspect important est la collecte du degré d'appréciation des utilisateurs pour une ressource donnée. Afin d'obtenir un plus grand nombre de notes, une fonction d'évaluation implicite basée sur la formule de Chan (Chan, 1999), en parallèle avec une fonction classique de collecte des évaluations (évaluation explicite), peut aussi être envisagée. Cette formule se charge d'estimer les évaluations que l'utilisateur est susceptible d'attribuer à une ressource donnée à partir de critères implicites. Cette méthode de collecte des préférences des utilisateurs permet de réduire le nombre d'évaluations manquantes et donc d'accroître la qualité des prédictions du système de recommandation.

Chapitre III

Système de recommandation distribué

« La centralisation, c'est l'apoplexie au centre, la paralysie aux extrémités. »

– Félicité de Lamennais (1782 – 1854)

3.1. Motivation

Depuis quelques années, les architectures distribuées gagnent énormément de terrain, tandis que les architectures centralisées passent petit à petit au second plan. Une des explications est le fait que l'architecture centralisée pose des problèmes de sécurité, de robustesse, de coût, et de bande passante (Androutsellis-Theotokis et Spinellis, 2004). Ces problèmes sont directement liés à l'utilisation d'un serveur central. Ainsi, l'avantage principal de l'architecture distribuée est l'absence d'une autorité centrale. En effet, chaque pair (utilisateur ou nœud) agit comme un serveur et un client. De ce fait, cette architecture préserve, en théorie, le total anonymat des utilisateurs. En évitant de communiquer des données personnelles au serveur, on échappe au problème d'identification de sécurité. De plus, dans une architecture décentralisée, la puissance de calcul réside au niveau du client. Ainsi, on n'a plus besoin de serveurs coûteux et d'une très grande bande passante. Enfin, cette architecture assure une meilleure disponibilité du service. Par exemple, dans le cadre d'un système de recommandation distribué, si un pair se déconnecte du réseau, ceci diminuera généralement de façon minime la qualité des recommandations. Dans le cas d'une architecture centralisée, si le serveur se déconnecte (tombe en panne), le fonctionnement global du système est affecté (Goth, 2006).

Cependant, si nous voulons nous départir du serveur central dans un système de recommandation, il faut trouver un moyen de trouver les utilisateurs du voisinage et puis de les faire communiquer. En effet, contrairement aux réseaux centralisés, où il suffisait de se connecter au serveur pour avoir accès à la liste des utilisateurs et ainsi aux informations, il faut aller les chercher. Dans ce but, il faut, en premier lieu, apprendre la topologie du réseau auquel l'utilisateur est connecté ; puis, il faut rechercher l'information sur un ensemble de nœuds ; enfin, il faut recevoir une réponse d'un ou plusieurs nœuds répondants aux critères. Le modèle de réseaux pair-à-pair entièrement décentralisé permet de concevoir des systèmes de très grande taille à forte disponibilité, tout cela à faible coût. Cependant, un inconvénient de ce réseau est qu'il est ouvert au parasitisme par la désinformation. En d'autres termes, la liberté de comportement de chaque pair, due à l'absence d'autorité centrale, peut engendrer des comportements de tromperie. Par exemple, dans les applications de partage de fichiers basées sur une architecture pair-à-pair, les utilisateurs peuvent fournir intentionnellement des documents erronés (Miller, Konstan et Riedl, 2004 ; Miller, 2003 ; Ziegler, 2005 ; Ziegler *et al.*, 2005). Les comportements de tromperie seront abordés, plus en détail, par la suite (chapitre 4 et chapitre 5).

Dans ce chapitre, nous allons, dans un premier temps, expliquer pourquoi le domaine des réseaux pair-à-pair suscite autant d'engouement. À cet effet, nous allons nous intéresser aux différents domaines d'application de ces réseaux. Ensuite, nous allons présenter, brièvement, les différentes classes de réseaux pair-à-pair, ainsi que les différentes techniques de repérage de l'information qui peuvent être utilisées dans un système de recommandation distribué. Enfin, nous allons passer en revue les différents modèles de système de recommandation distribué rencontrés dans la littérature.

3.2. Domaines d'application des réseaux pair-à-pair

Le domaine des réseaux pair-à-pair est un sujet d'actualité. Pour comprendre l'intérêt que suscitent ces réseaux, le plus simple est de voir comment, depuis quelques années, différentes applications les exploitent. Dans la littérature, nous pouvons distinguer cinq classes principales d'applications pair-à-pair (Androutsellis-Theotokis et Spinellis, 2004 ; Goth, 2006):

Le calcul distribué (« Grid Computing »). Le réseau pair-à-pair peut être mis à profit pour exploiter en parallèle les capacités de calcul d'un grand nombre d'ordinateurs. Ainsi, ce domaine d'application regroupe tous les systèmes capables de partager les ressources des ordinateurs à travers le réseau intranet ou internet. Généralement, ces systèmes sont utilisés dans des applications qui nécessitent une grande puissance de calcul. En effet, de nombreux projets scientifiques exploitent ces systèmes. Ces derniers se présentent, généralement, sous la forme d'économiseurs d'écran qui lorsque votre ordinateur n'est pas employé, utilisent la puissance de calcul de votre ordinateur et celui des autres pour traiter des données. Par exemple, *Seti@home* (<http://setiathome.ssl.berkeley.edu>) utilise ces systèmes pour la recherche de vie extraterrestre ; *Genome@home* (<http://genomeathome.stanford.edu>) les utilise pour le décryptage du génome humain ; *Folding@home* (<http://folding.stanford.edu>) les exploite pour le repliement des protéines ; *Data Grid Computing* (www.eu-datagrid.org) les utilise pour la physique des hautes énergies ; *Parabon* (www.parabon.com) et *Entropia* (www.entropia.com) vendent l'accès à leur système de calcul distribué pour différents types de projets, etc.

Le partage de contenu. La technologie pair-à-pair exploite le contenu réparti sur le réseau pour mettre en œuvre une application de gestion de connaissance. Plus précisément, ces applications permettent d'établir des relations d'échanges réciproques de contenus entre des ordinateurs distants reliés au réseau intranet ou internet. L'exemple le plus connu est les systèmes de partage de musique et de vidéo via internet, comme *Napster*, *Kazaa*, *eMule*, *LimeWire*, *µTorrent*, etc.

Le travail collaboratif. Ce type d'application est principalement réservé aux entreprises. Le logiciel le plus connu dans ce domaine est *Groove*. Ce dernier fournit aux utilisateurs des mécanismes de dialogue en temps réel (texte, audio et vidéo), de partage de documents ou d'outils, de synchronisation de données en temps réel, et de gestion de la sécurité (authentification et cryptage). Il existe d'autres applications qui offrent le même service, comme *NetMeeting*, *Bleu* et *Magi*.

La communication. Plusieurs services de communication utilisent les réseaux pair-à-pair, comme le clavardage (*ICQ*, *Jabber*), la téléphonie IP, la vidéoconférence (*NetMeeting*),

etc. L'exemple le plus connu est l'application *Skype* qui offre en se basant sur une architecture pair-à-pair des services de clavardage, de téléphonie IP, et de vidéoconférence.

D'autres types d'application. La puissance des réseaux pair-à-pair est aussi utilisée dans plusieurs autres domaines d'application, comme les systèmes de base de données (*PIER*, *PIAZZA*, *EDUTELLA*), les jeux en ligne, la recherche distribuée d'informations (*OpenCola*, *Copernic*), *P2P TV*² (*TVants*, *TVUPlayer*, *SopCast*), *P2P Radio*³ (*PeerCast*, *P2P-Radio*, *Streamerp2p*), etc.

3.3. Classes de réseaux pair-à-pair

Dans cette section, nous essayons de présenter l'ensemble des structures de réseaux pair-à-pair existantes. Dans la littérature, on retrouve trois types de structures de réseaux pair-à-pair (Androutsellis-Theotokis et Spinellis, 2004 ; Lua *et al.*, 2005):

- ✓ Les réseaux pair-à-pair hybrides, aussi appelés réseaux pair-à-pair centralisés, ne sont pas des réseaux pair-à-pair purs. En effet, ces réseaux maintiennent un index, de tous les documents partagés par les pairs participants, dans un ou plusieurs serveurs centraux (aussi appelés serveurs d'annuaire). En revanche, les transferts de données se passent directement entre les pairs, sans faire intervenir ces derniers. Par exemple, *Napster*, dont la topologie est illustrée par la figure 3.1, repose sur un unique serveur d'annuaire. Par contre, *Kazaa*, dont la topologie est illustrée par la figure 3.2, étend le système centralisé de *Napster* à un serveur lui-même distribué. Pour ce faire, *Kazaa* introduit le concept de *Super-pair*. Ainsi, chaque pair est relié un *Super-pair*, où il tient l'index des fichiers qu'il partage et où il envoie toutes ses requêtes. Par la suite, les *Super-pairs* communiquent entre eux pour mettre à jour l'index global. Les réseaux pair-à-pair hybrides sont rapides et offrent la garantie de trouver tous les résultats

² *P2P TV* fait référence à un logiciel qui permet de redistribuer des flux vidéo en temps réel sur le réseau pair-à-pair

³ *P2P TV* fait référence à un logiciel qui permet de redistribuer des flux audio en temps réel sur le réseau pair-à-pair

possibles. Cependant, ces réseaux sont trop coûteux (serveur central) et ne sont pas fiables (perte de service si le serveur tombe en panne). En outre, les résultats de recherche dans ces réseaux comportent un nombre considérable de liens morts en fonction de la fréquence de mise à jour de l'index des utilisateurs (Liang, Kumar et Ross, 2004).

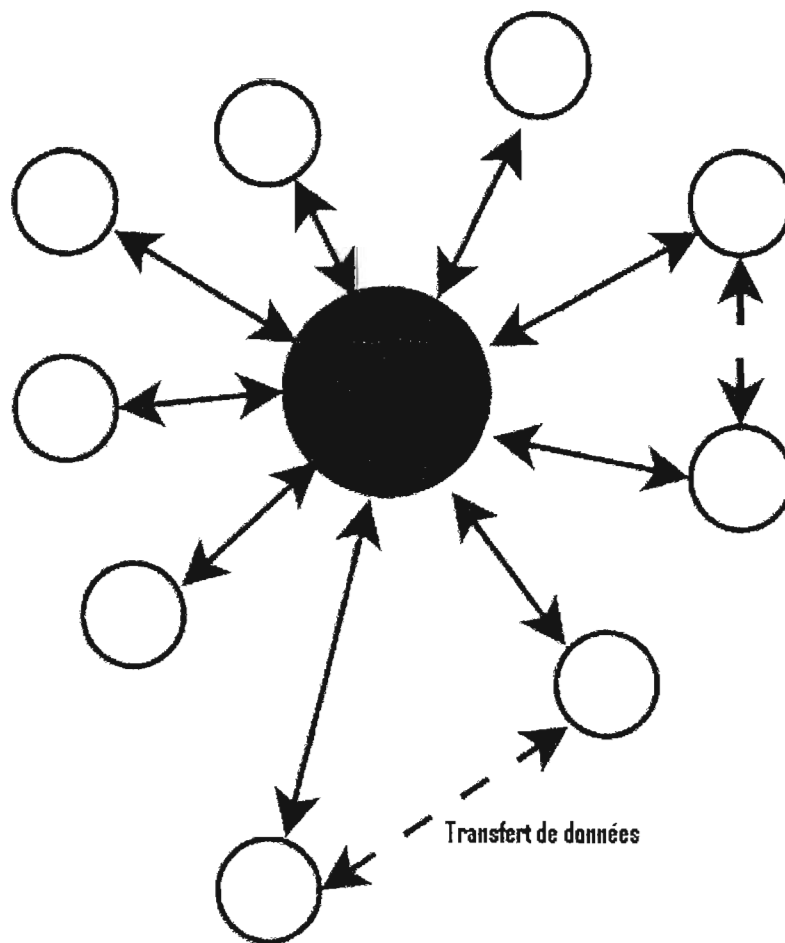


Figure 3.1 Topologie du réseau *Napster*

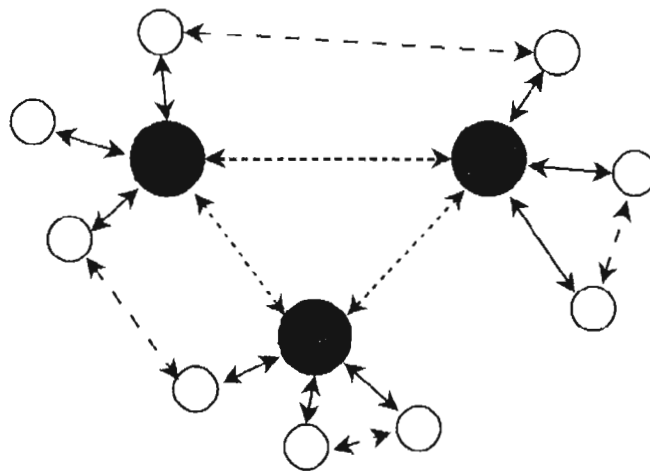


Figure 3.2 Topologie du réseau *Kazaa*

- ✓ Les réseaux pair-à-pair non structurés, aussi qualifiés de réseaux faiblement contrôlés, se focalisent essentiellement sur le partage des données et non sur les règles de leur stockage. En effet, le stockage des données est très faiblement contrôlé et les pairs (utilisateurs) sont incapables de décider quelles données ils veulent stocker et partager avec les autres pairs du réseau. Ce problème réduit considérablement la persistance des données et ainsi leur disponibilité. La figure 3.3 illustre une représentation symbolique des réseaux pair-à-pair non structurés, comme *Gnutella*, *Gia*, *Freenet*, etc.

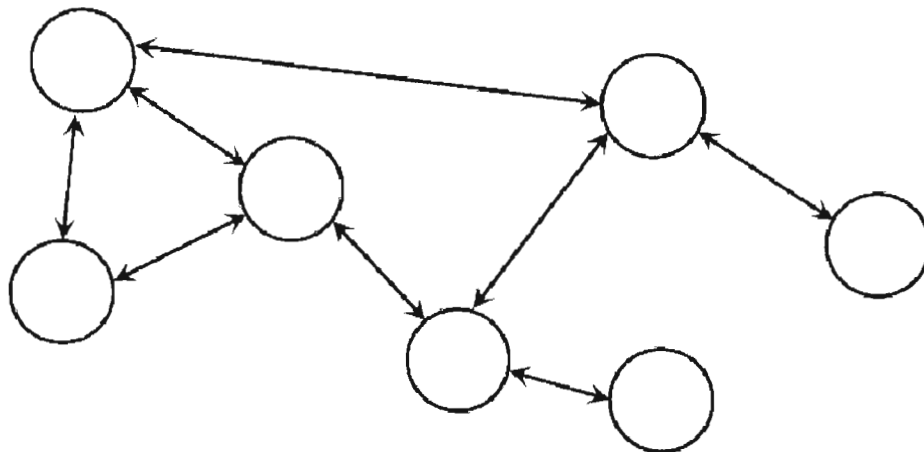


Figure 3.3 Représentation symbolique des réseaux pair-à-pair non structurés

- ✓ Les réseaux pair-à-pair structurés, aussi appelés réseaux fortement contrôlés, sont généralement vus, en opposition aux réseaux pair-à-pair non structurés, comme des réseaux imposant un contrôle plus strict sur le stockage des données et la topologie du réseau pair-à-pair (Voir figure 3.4). Dans ce type de réseaux comme *Chord*, *Can*, *Tapestry*, etc., les pairs sont capables de définir quelles données ils veulent stocker et partager. Les réseaux pair-à-pair structurés reposent, généralement, sur le principe d'une table de hachage distribuée (*DHT* : *Distributed Hash Table*). Une *DHT*, illustrée par la figure 3.5, est une fonction distribuée qui, à partir d'un hash, rend une valeur. Plus précisément, le *DHT* est une structure de données qui associe une clé, appelée *hash*, à chaque ressource de réseau, et ainsi fournit un accès rapide à ces dernières. À cet effet, les deux principales fonctions fournies sont *insert(hash, data)* qui insère une donnée dans la *DHT* et *lookup(hash)* qui retourne la donnée stockée en *hash*.

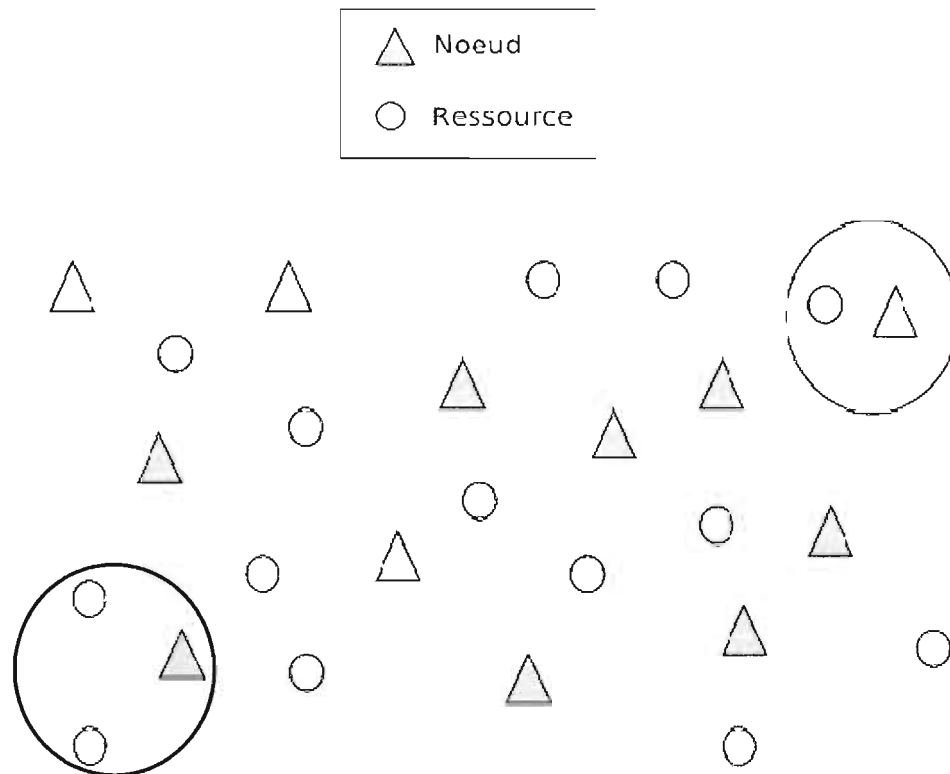


Figure 3.4 Représentation symbolique du DHT

3.4. Techniques de repérage de l'information pour les réseaux pair-à-pair

Dans cette section, nous présentons l'ensemble des techniques de repérage de l'information dans les réseaux pair-à-pair. L'objectif est de donner une vue des différents mécanismes qui peuvent être utilisés dans un système de recommandation distribué (Androutsellis-Theotokis et Spinellis, 2004).



Figure 3.5 Table de hachage distribuée

3.4.1. Approches centralisées

Le système pair-à-pair centralisé utilise une base de données centrale où chaque pair fournit un index de tous ses documents partagés. Un pair qui recherche une information doit automatiquement passer par un serveur central pour *Napster* ou un des *Super-Pairs* pour *Kazaa*. En recevant la requête de recherche, ces derniers recherchent dans leur base de données préalablement établie (Liang, Kumar et Ross, 2004). Ces étapes sont illustrées par la figure 3.6.

3.4.2. Approches décentralisées non structurées

Il existe dans ce type de réseaux deux techniques de recherche. D'une part, les algorithmes de recherche à l'aveugle qui n'emploient aucune technique explicite pour guider la recherche, d'autre part, les algorithmes de recherche dirigée qui utilisent des techniques

spécifiques pour guider la recherche (Androutsellis-Theotokis et Spinellis, 2004 ; Lua *et al.*, 2005).

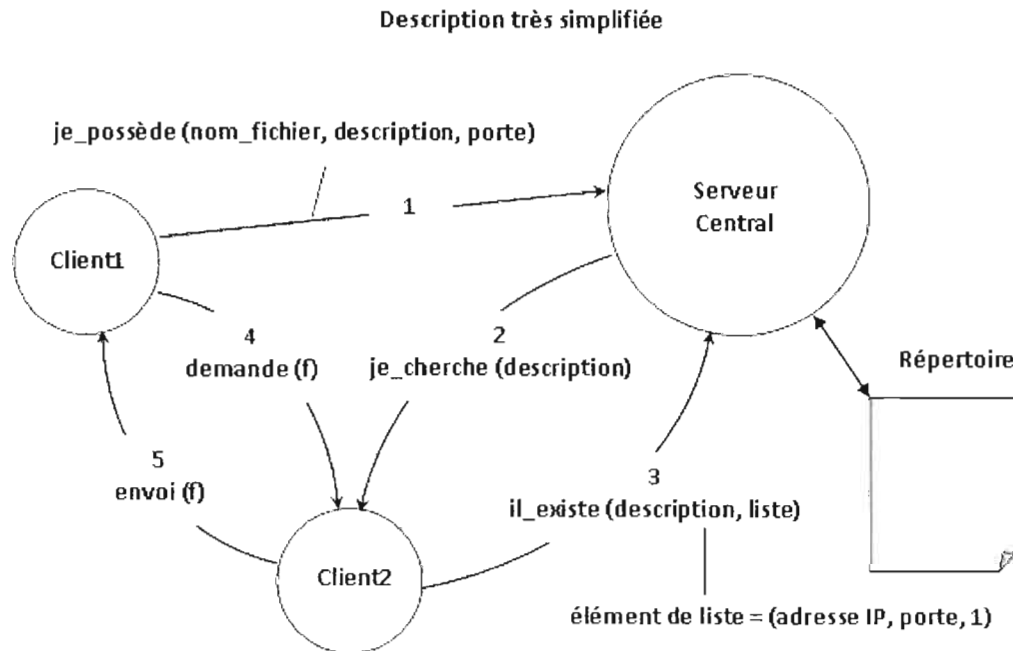


Figure 3.6 Approche centralisée

3.4.2.1. Techniques de recherche à l'aveugle

Recherche en largeur (BFS : « Breadth First Search »). Le *BFS*, illustré par la figure 3.7, est une technique de recherche largement répandue dans les réseaux pair-à-pair, comme *Gnutella*. Il est considéré comme une approche naïve. En effet, un pair produit un message de requête qui est propagé à tous ses voisins (pairs). Puis, quand un pair reçoit cette requête, d'abord, il la fait suivre à tous les pairs de son voisinage, autre que l'expéditeur. Ensuite, il recherche dans sa base de données locale les résultats appropriés. Enfin, si un pair a un résultat, alors il produit un message de *QueryHit* pour transmettre le résultat. Le message de *QueryHit* inclut aussi des informations telles que le nombre de documents et de la connectivité du pair. Comme nous pouvons le remarquer, cette technique est basée sur le principe d'inondation du réseau entier pour la recherche d'objet. En effet, une requête est propagée le long de tous les liens et entre en contact avec tous les pairs accessibles. Bien que

ce soit une technique simple à mettre en œuvre, elle pose des problèmes de performance et de surcharge du réseau (congestion du réseau). Il est à noter qu'il existe une technique pour éviter l'inondation totale du réseau avec des messages. Cette technique associe à chaque requête un paramètre appelé *Time-To-Live (TTL)* qui détermine le nombre maximum de sauts qu'une requête donnée devrait faire (Kalogeraki, Gunopulos et Zeinalipour-Yazti, 2002).

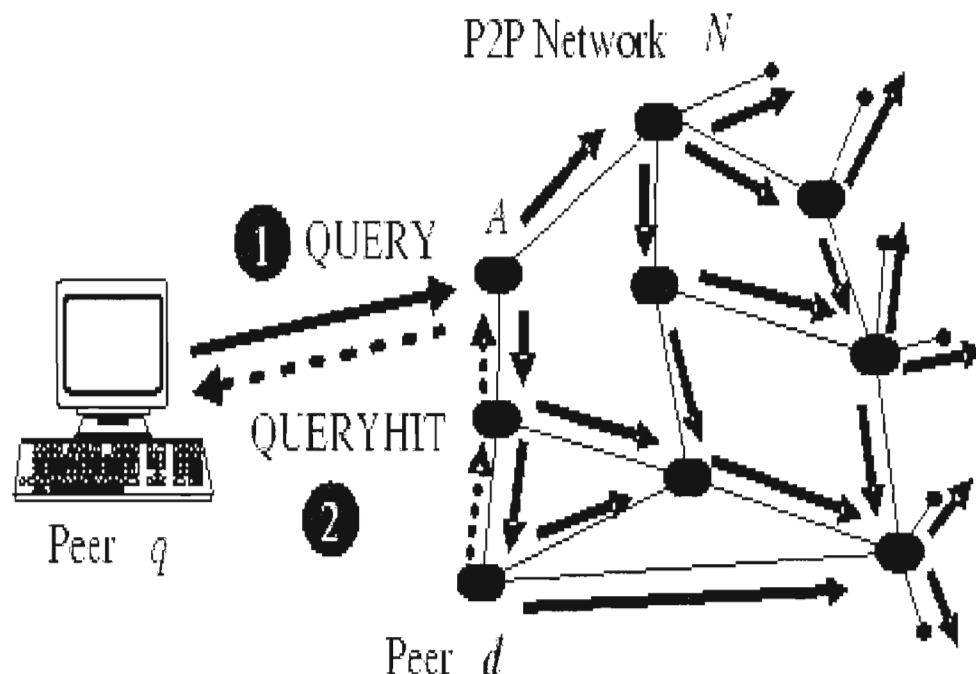


Figure 3.7 Technique *BFS* (Zeinalipour-Yazti, Kalogeraki et Gunopulos, 2004)

Recherche aléatoire en largeur (RBFS : «Random Breadth-First-Search »). L'idée de cet algorithme, illustré par la figure 3.8, est de modifier l'algorithme original du *BFS* de sorte que le message de requête est envoyé seulement à un nombre limité de voisins. À cet effet, un pair fait suivre la requête seulement à une fraction de ses pairs, choisie au hasard. Le pourcentage des pairs sollicités est un paramètre de l'algorithme. L'avantage du *RBFS* est qu'il n'exige pas la connaissance globale. En effet, un pair peut prendre des décisions locales d'une façon rapide puisqu'il doit seulement choisir une partie de ses pairs. Cependant, bien que cet algorithme réduise le nombre de messages, il reste peu fiable. En effet, avec cette

méthode, des parties du réseau peuvent devenir inaccessibles (Kalogeraki, Gunopulos et Zeinalipour-Yazti, 2002).

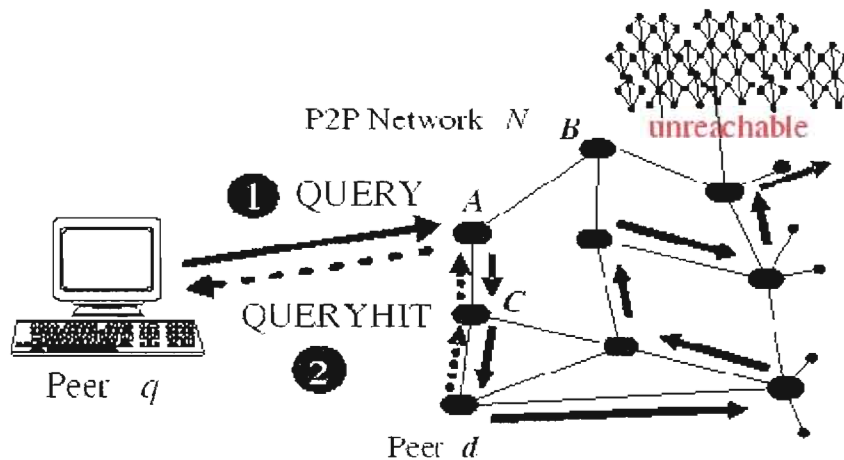


Figure 3.8 Technique *RBFS* (Zeinalipour-Yazti, Kalogeraki et Gunopulos, 2004)

Technique « *Random Walkers* ». L'idée principale derrière cette technique, illustrée par la figure 3.9, est que chaque pair propage aléatoirement le message de requête à un nombre K de ses voisins. Chacun de ces pairs fait suivre la requête à n'importe lequel de ces voisins (aléatoirement). Ces messages de requêtes sont désignés sous le nom de *Walkers*. L'avantage de cette approche est qu'elle produit un nombre de messages qui est indépendant de la topologie du réseau : $K * TTL$. Bien que cet algorithme ressemble à celui du *RBFS*, dans ce dernier, l'augmentation du nombre de messages utilisés est exponentielle tandis que dans le modèle de *Random Walkers* cette augmentation est linéaire. Cependant, il reste le problème du choix aléatoire des pairs qui n'assure pas toujours un résultat final (Lv *et al.*, 2002).

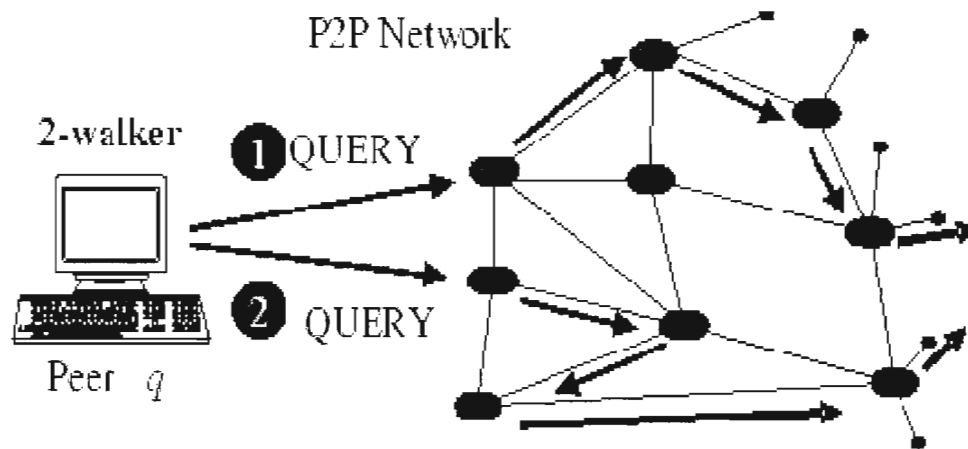


Figure 3.9 Technique « *Random Walkers* » (Zeinalipour-Yazti, Kalogeraki et Gunopulos, 2004)

3.4.2.2. Techniques de recherche dirigée

Technique « *Most Results in Past* » (>RES). Dans cette technique illustrée par la figure 3.10, chaque pair fait suivre une requête à un sous-ensemble de ses voisins choisis sur la base de certaines statistiques. Plus précisément, cet algorithme utilise le nombre de résultats des requêtes antérieures retournées par le pair comme heuristique pour le choix du sous-ensemble de pairs. Dans ce but, la requête est envoyée à un nombre K de pairs qui ont retourné le plus grand nombre de résultats pour les M dernières requêtes, M et K étant des paramètres de l'algorithme (Yang et Garcia-Molina, 2001).

Technique « *Adaptive Probabilistic Search* » (APS). C'est une technique semblable à l'algorithme du *Random Walkers*. Dans cet algorithme, chaque pair déploie un index local, qui calcule la probabilité relative de chaque voisin à être choisi comme prochain saut pour une future requête. En d'autres termes, la différence principale avec l'algorithme du *Random Walkers* est que dans l'algorithme APS, le pair utilise un retour d'information (*feedback*) sur les recherches précédentes pour choisir, grâce aux probabilités, les futurs pairs, plutôt qu'un choix aléatoire. De ce fait, l'algorithme APS offre de meilleurs résultats que l'algorithme du *Random Walkers* (Tsoumakos et Roussopoulos, 2003).

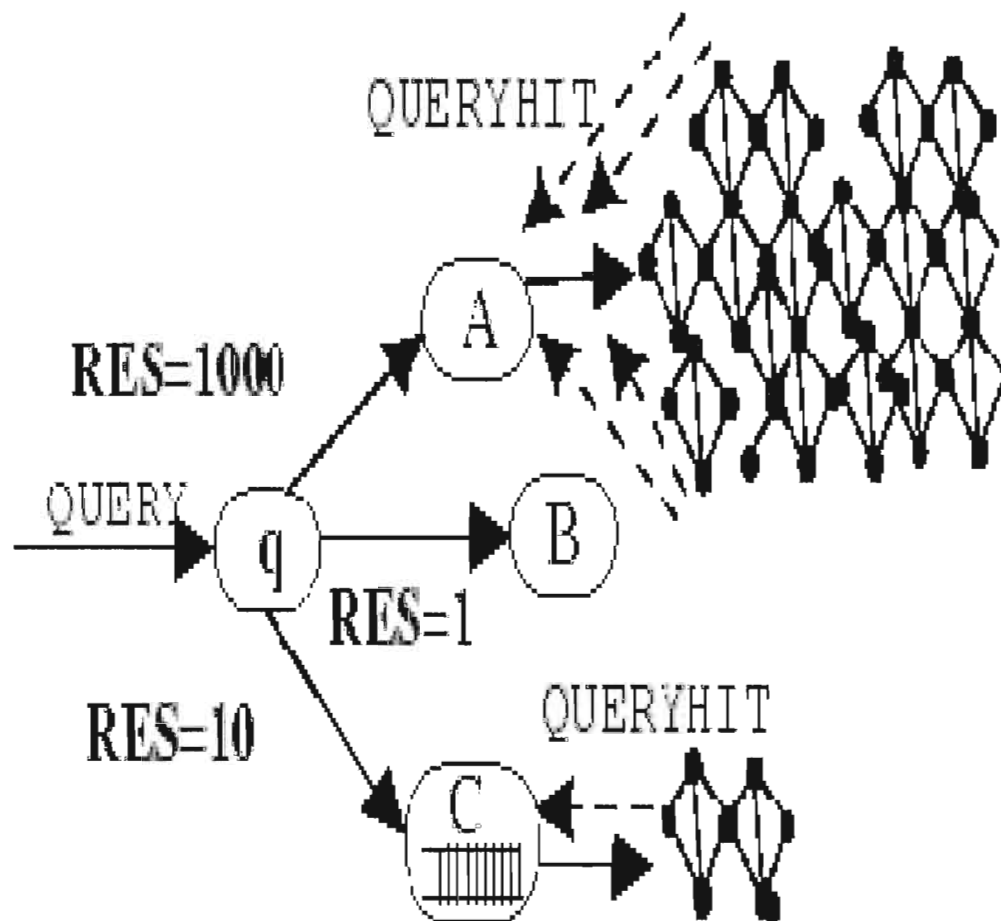


Figure 3.10 Technique $>RES$ (Zeinalipour-Yazti, Kalogeraki et Gunopulos, 2004)

Recherche en profondeur (« Depth-First-Search »). *Freenet* est un réseau non structuré ou peu structuré. Chaque nœud (pair) du réseau dispose d'une clé unique (le hash *SHA-1*), de même que chaque ressource (fichier). Lors d'une requête, le nœud envoie la requête au nœud connu ayant l'identifiant le plus proche de l'identifiant de la requête. Ce nœud fait de même jusqu'à trouver un nœud possédant l'information répondant à la requête. À cet effet, l'algorithme calcule la distance lexicographique entre les clefs. Un mécanisme de mise en cache des réponses permet aux nœuds de se spécialiser dans les requêtes qui leur sont dirigées (Clarke *et al.*, 2000). Les étapes de cet algorithme sont présentées dans la figure 3.11.

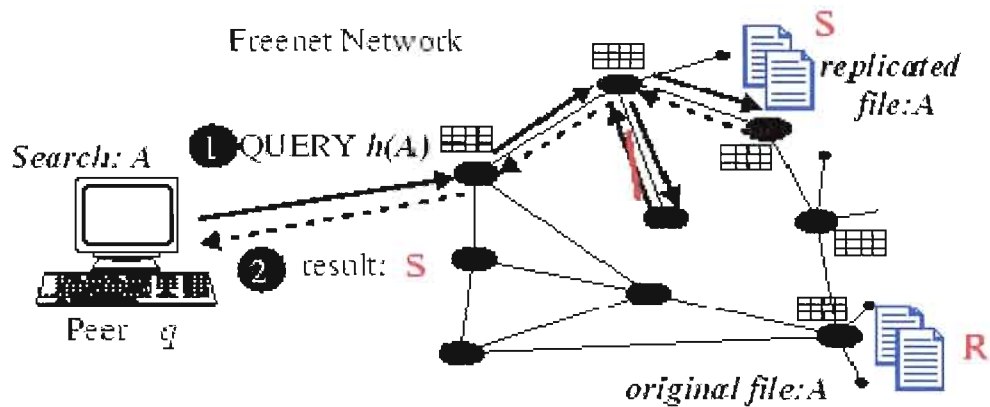


Figure 3.11 Recherche en profondeur (Zeinalipour-Yazti, Kalogeraki et Gunopulos, 2004)

3.4.3. Approches décentralisées structurées

Technique « Consistent Hashing ». Le réseau *Chord* repose sur une structure en anneau, illustrée par la figure 3.12. Chaque nœud du réseau dispose d'une clé unique (le hash *SHA-1*), de même que chaque ressource. Chaque nœud doit connaître son successeur et chaque ressource est affectée à un nœud. Pour ce faire, une ressource avec une clé *Key1* est affectée au nœud possédant le premier identifiant supérieur à *Key1*. De ce fait, un nœud qui se connecte au réseau peut prendre le contrôle d'une ressource existante. On parcourt les nœuds séquentiellement jusqu'à trouver la paire de nœuds qui encadre la clé recherchée. Ainsi, le dernier est le nœud cherché. De même, une table de pointeurs (table de raccourcis vers d'autres nœuds), illustrée par la figure 3.13, peut être utilisée pour optimiser la recherche. De plus, chaque nœud interroge périodiquement son successeur pour déterminer le prédécesseur de ce dernier, ce qui permet de détecter une insertion (Stoica *et al.*, 2001). Le réseau *CAN* (« *Content Addressable Network* ») utilise un système virtuel de coordonnées cartésiennes. En d'autres termes, il partitionne l'espace des clés (le hash *SHA-1*) parmi tous les nœuds (voir figure 3.14). Ainsi, chaque nœud a un morceau d'espace qui lui est associé. Par contre, le réseau *Tapestry* utilise une technique de routage de proche en proche dans l'espace des clés (le hash *SHA-1*) par suffixe commun. La figure 3.15 illustre le routage ($3745 \rightarrow 3BA8$) = ($???? \rightarrow ??A8 \rightarrow ?BA8 \rightarrow 3BA8$) (Androutsellis-Theotokis et Spinellis, 2004).

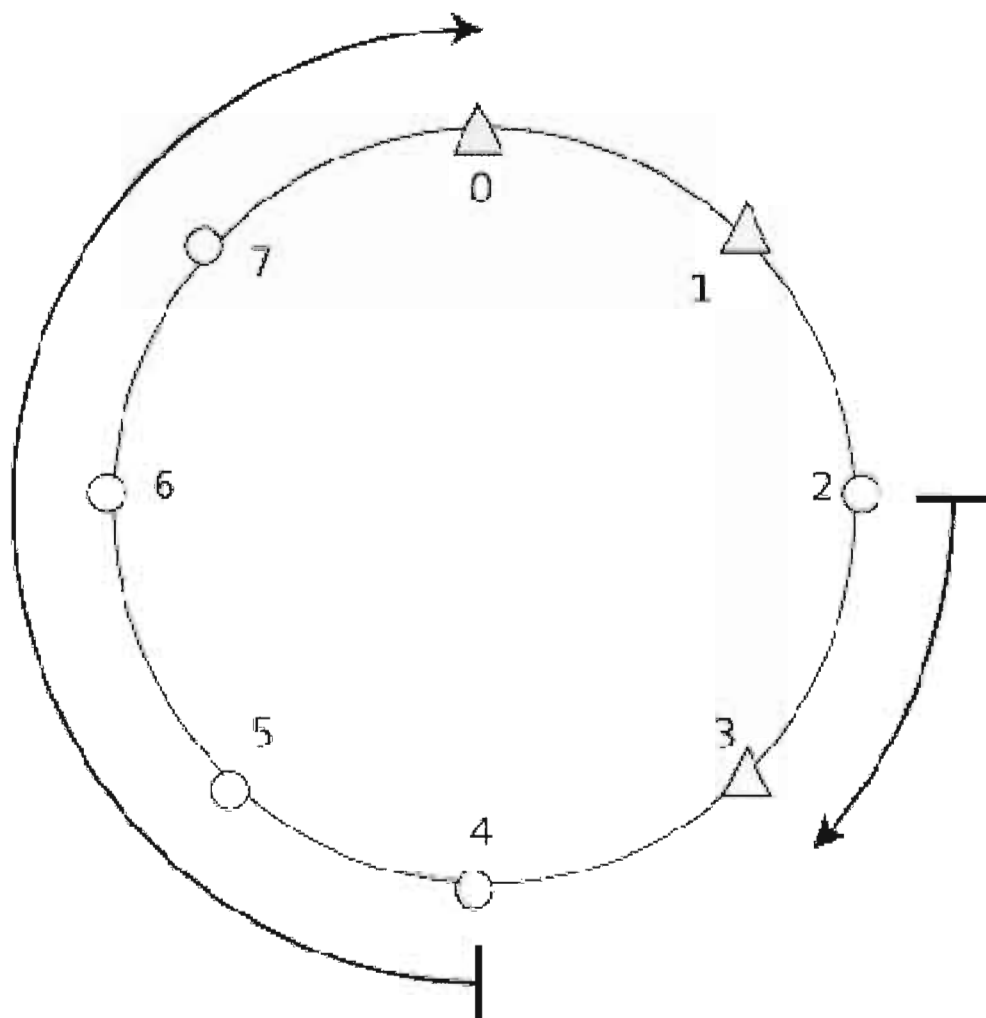


Figure 3.12 Chord

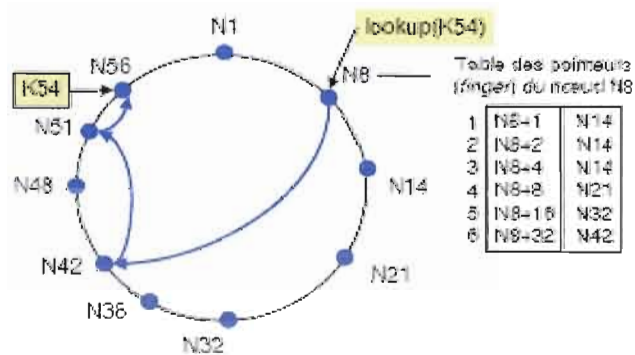


Figure 3.13 Table des pointeurs

SETS (« *Search Enhanced by Topic segmentation* »). *SETS* est une architecture pour la recherche efficace dans les réseaux pair-à-pair. Dans ce but, le système arrange les nœuds participants en segments thématiques à travers le réseau. En effet, la philosophie fondamentale de *SETS* est d'arranger des nœuds (pairs) de manière à ce qu'une requête de recherche interroge uniquement un petit sous-ensemble de nœuds où la plupart des documents pertinents sont localisés. Plus précisément, *SETS* partitionne les nœuds en des segments thématiques, où les documents semblables appartiennent au même segment. Chaque segment thématique a une description succincte appelée *Topic centroid*. Ces derniers sont employés pour choisir un petit ensemble approprié de segments thématiques. Ainsi, les nœuds sont placés dans un réseau segmenté qui se compose de deux types de liens, 1) des liens de longue distance reliant les nœuds des différents segments ; 2) des liens de courte distance qui relient des nœuds dans le même segment. Une requête, à un segment particulier, se propage en deux étapes : d'abord, la requête est conduite le long des liens de longue distance pour atteindre un emplacement aléatoire appartenant au segment cible. Ensuite, les liens de courte distance sont employés pour propager la requête aux nœuds dans le segment. Chacun des nœuds dans le segment fait une recherche dans sa base de données et les résultats de tous les nœuds sont combinés et sont fournis au pair qui a émis la requête (Zeinalipour-Yazti, Kalogeraki et Gunopulos, 2004).

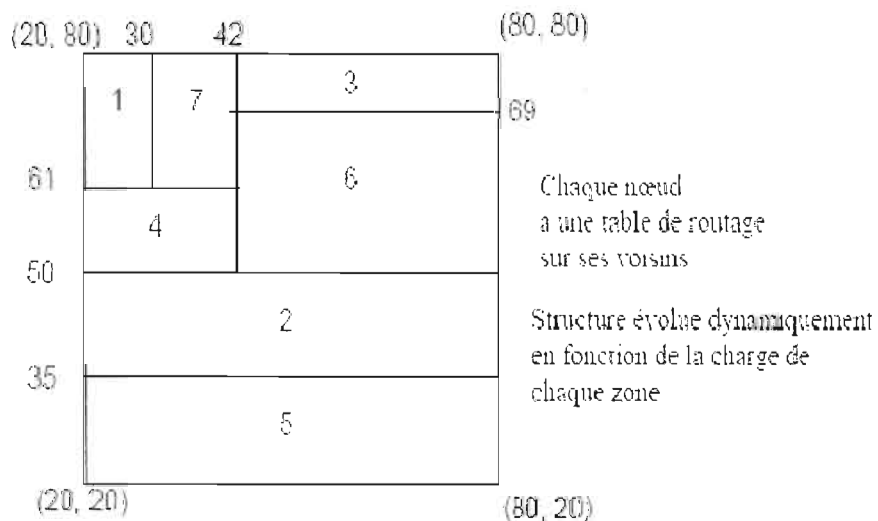


Figure 3.14 CAN

Mécanisme intelligent de recherche (ISM : « *Intelligent Search Mechanism* »).

L'objectif de cet algorithme, illustré par la figure 3.16, est de repérer l'information d'une manière plus rapide et plus efficace. À cet effet, *ISM* utilise un profil, au niveau de chaque pair, représentant les dernières réponses obtenues de chacun des voisins et un « *RelevanceRank* » représentant le rang du pair calculé en fonction du profil. À chaque requête de recherche, l'algorithme sélectionne un nombre limité de voisins, à qui il envoie la requête de recherche, en fonction du profil et du « *RelevanceRank* ». Le pair suivant fait de même jusqu'à trouver un nœud possédant l'information répondant à la requête (Zeinalipour-Yazti, Kalogeraki et Gunopulos, 2004).

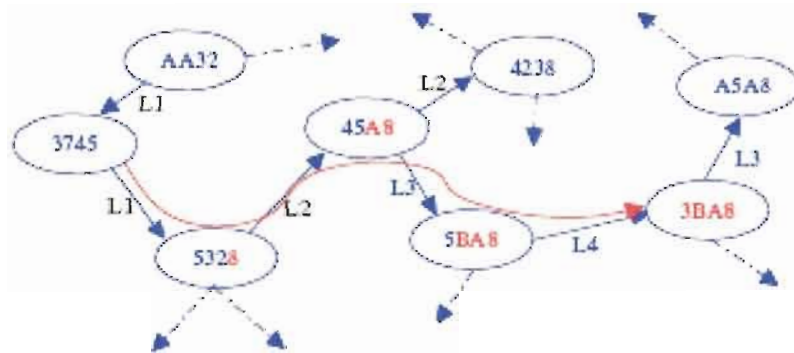


Figure 3.15 Tapestry

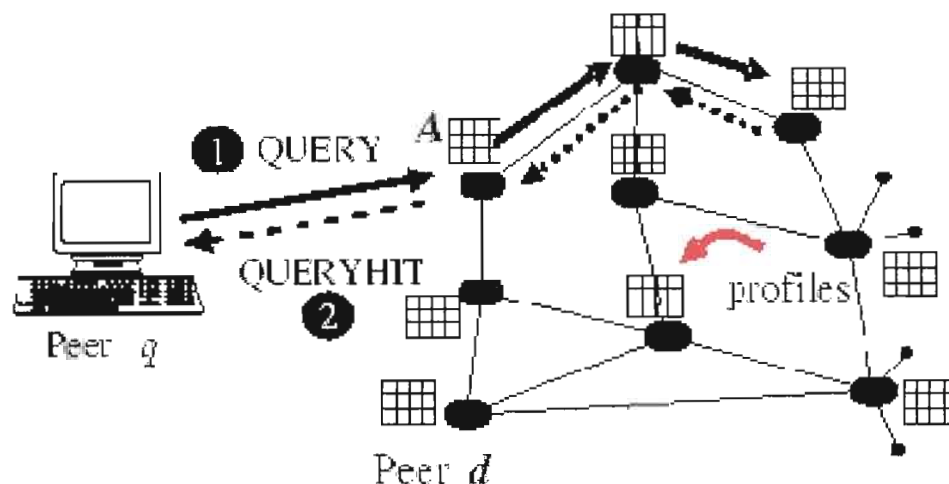


Figure 3.16 Technique ISM (Zeinalipour-Yazti, Kalogeraki et Gunopulos, 2004)

Keyword Relationship. Ce système est un algorithme de recherche de documents par mots clés dans un réseau pair-à-pair. Cet algorithme améliore le repérage de l'information désirée par expansion de la requête. En d'autres termes, le système ajoute des mots-clés appropriés à la requête originale. Cette expansion de la requête est basée sur une base de données, appelée *KRDB*, de relations entre mots clés. Cette dernière est contrôlée et mise à jour par les pairs participants dans un mode distribué (Nakauchi *et al.*, 2003).

Swarm Search. Cet algorithme de recherche est basé sur l'imitation du comportement de colonies de fourmis. Il emploie un nombre élevé d'agents avec des comportements simples, et génère une espèce d'agents (type d'agents) pour chaque requête utilisateur (spécialisation). Ainsi, les agents de la même espèce ont l'unique mission de rechercher le chemin le plus court aux ressources correspondant aux critères donnés par une requête de l'utilisateur (Wittner, Heegaard et Helvik, 2003).

3.5. Systèmes de recommandation distribués

Avec quelques exceptions (Canny, 2002a ; Han *et al.*, 2004 ; Kong *et al.*, 2006 ; Lathia, Hailes et Capra, 2007 ; Memmi et Nérot, 2003 ; Miller, Konstan et Riedl, 2004 ; Oka, Morikawa et Aoayama, 2004 ; Olsson, 2003 ; Pingfeng *et al.*, 2007 ; Wang *et al.*, 2006b ; Wang *et al.*, 2006a ; Zaier, Godin et Faucher, 2008b), les systèmes de recommandations sont dans la majeure partie des cas centralisés (Miller, Konstan et Riedl, 2004). En d'autres termes, ces outils sont basés sur un système de calcul de prédictions centralisé et une base de données centralisée. Cependant, comme nous l'avons déjà remarqué, ces dix dernières années, les systèmes de recommandation distribués sont devenus de plus en plus populaires. En effet, la première recherche sur ce domaine était probablement le système Yenta (Foner, 1997, 1999). Ce dernier, basé sur l'approche multi-agents, est le premier système de recommandation distribué. En se basant sur les travaux de Foner, Olsson (1998, 2003) propose un autre système de recommandation distribué, basé sur une technique de regroupement. Vu le nombre considérable de messages échangés entre les pairs, ces deux systèmes souffraient d'un problème de congestion du réseau.

Mui et *al.* (2001) proposent un système de recommandation distribué basé sur la réputation de l'utilisateur, plutôt que la similarité entre utilisateurs. Suivant le même principe, Kinatader et Rothermel (2003) introduisent une architecture, semblable à celles de Foner et d'Olsson, basée sur la réputation des utilisateurs. Montaner (2003) utilise la confiance plutôt que la réputation dans son système de recommandation distribué. Massa et Avesani (2004) présentent un algorithme de recommandation distribué, basé sur une simple technique de propagation de la confiance.

Memmi et Nérot (2003) proposent un algorithme distribué pour la recommandation de documents, basé sur les réseaux sociaux. De plus, cet algorithme introduit le concept de profils multiples pour un unique utilisateur. Han et *al.* (2004) proposent une heuristique, semblable à la table de hachage distribuée (*DHT*), pour chercher les utilisateurs semblables dans son système de recommandation distribué. Miller et *al.* (2004) explorent cinq techniques de repérage de l'information dans les réseaux pair-à-pair, pour différents systèmes de recommandation distribués, à savoir un serveur central, *Random Discovery* similaire à la technique utilisée par Gnutella, *Transitive Traversal*, table de hachage distribuée (*DHT*), et *Secure Blackboard*.

Kong et *al.* (2006) proposent un système distribué pour le filtrage de courriels. Wang et *al.* (Wang et *al.*, 2006a) présentent un algorithme de recommandation distribué pour les systèmes de partage de fichiers. Wang et *al.* (2006b) proposent aussi un algorithme de recommandation distribué pour les systèmes de *P2P TV*. Pingfeng et *al.* (2007) introduisent un algorithme de recommandation distribué basé sur le concept de connaissance. Zaier et *al.* (2008b) proposent un système de recommandation distribué basé sur un algorithme de discrimination du voisinage. Lam et *al.* (2006) passent en revue les différents problèmes de protection de la vie privée dans un système de recommandation distribué. En se basant sur cette étude, Lathia et *al.* (2007) proposent un modèle de système de recommandation distribué protégeant la vie privée des utilisateurs.

3.6. Conclusion

Ces dix dernières années, les systèmes de recommandation distribués sont devenus de plus en plus populaires. Ces systèmes permettent de distribuer le procédé de filtrage collaboratif tant en terme de contenu que de calculs. Ainsi, l'algorithme résultant est capable de fournir des recommandations en temps réel et de traiter un nombre considérable de profils avec des ressources limitées. Il permet également de prendre en compte la protection de la vie privée des usagers. Cependant, la distribution des procédés de filtrage collaboratif pose un nouveau problème, notamment en termes de localisation des données, puisqu'il n'y a plus d'autorité centralisée pour localiser les données. Néanmoins, il existe différents mécanismes de repérage d'information qui peuvent être utilisés dans un système de recommandation distribué. Une étude comparative des performances de différentes techniques de repérage d'information dans le cadre d'un système de recommandation est présentée par Miller et *al.* (2004).

Chapitre IV

Agent social de recommandation

« L'homme est un animal social. »

- Aristote (-384 - -322)

4.1. Introduction

Le présent chapitre s'intéresse au cadre de recherche sur les systèmes multi-agents dans les systèmes de recommandation. Plus précisément, ces systèmes de recommandation considèrent le système de filtrage collaboratif comme un ensemble d'agents indépendants, un agent pour chaque utilisateur du système. Le système est constitué d'une population d'agents autonomes en interaction, chaque agent possédant des capacités de communication, de coordination et de collaboration (Jacobsson, Rost et Holmquist, 2006). En fait, il existe un lien naturel entre les systèmes pair-à-pair et les systèmes multi-agents. En effet, la majorité des études sur les systèmes de recommandation distribués, présentées dans la section 3.5, tentent de combiner la technologie pair-à-pair et les systèmes multi-agents (Foner, 1997, 1999 ; Kinatader et Rothermel, 2003 ; Olsson, 1998, 2003).

Le domaine des systèmes multi-agents, s'il n'est pas récent, est actuellement un champ de recherche très actif. Le concept d'agent a été l'objet d'études depuis plusieurs décennies dans différentes disciplines. Il a été utilisé dans les systèmes à base de connaissances, la robotique, l'étude du langage naturel et d'autres domaines de l'intelligence artificielle, mais aussi dans d'autres disciplines comme la philosophie et la psychologie. Dans ces différentes disciplines, on trouve une multitude de définitions d'agents qui se ressemblent toutes, mais diffèrent selon le type d'application pour laquelle est conçu l'agent. Parmi ces différentes

définitions, Ferber (2007) propose une définition qui reflète les propriétés que doit posséder minimalement un agent :

« Un agent est une entité autonome, réelle ou abstraite, qui est capable d'agir sur elle-même et sur son environnement, qui, dans un univers multi-agents, peut communiquer avec d'autres agents, et dont le comportement est la conséquence de ses observations, de ses connaissances et des interactions avec les autres agents. »

Il ressort de cette définition des propriétés clés, telles que l'autonomie, l'action, la perception, la communication et l'aspect social. Ces notions sont définies comme suit (Zaier, 2002):

- ✓ Autonome : l'agent est capable d'agir sans l'intervention d'un tiers (humain ou agent) et contrôle ses propres actions ainsi que son état interne ;
- ✓ Action : l'agent intervient en agissant sur ses ressources propres ou sur son environnement direct ;
- ✓ Perception : l'agent doit être capable de percevoir son environnement à travers un ensemble de capteurs ;
- ✓ Communication : l'agent doit être capable de dialoguer avec les autres entités de son environnement ;
- ✓ Social : l'agent doit être capable, quand la situation l'exige, d'interagir avec les autres agents (logiciels et humains) afin d'achever ses tâches ou d'aider des agents à accomplir les leurs.

L'aspect social, thème central de ce chapitre, permet de résoudre l'un des problèmes majeurs des systèmes multi-agents : celui du contrôle. Ce dernier est donc établi par la collectivité d'agents et non de manière centralisée. Il faut donc que les agents parviennent à se coordonner, collaborer, gérer l'accès aux ressources, résoudre les conflits, échanger des informations en fonction du but à atteindre, et ce, en dépit du fait qu'il n'y ait pas de structure de contrôle centralisée. Dans ce but, deux types d'agents ont vu le jour : 1) Les

agents « cognitifs » qui consistent en des algorithmes basés sur des dispositifs cognitifs sophistiqués (inférences, représentations symboliques, négociation explicite avec les autres agents). Ces agents sont complexes et souvent destinés à des environnements préétablis (Castelfranchi, 1990). 2) Les agents dits « réactifs », quant à eux, ont montré leur capacité à résoudre des problèmes complexes (tris, exploration, puzzles, construction) en utilisant, non pas des compétences individuelles développées, mais avant tout la richesse des interactions avec leurs congénères et avec un environnement dynamique et non prévisible (Parunak, 1997).

Dans le reste de ce chapitre, nous présentons, dans la première partie, un rappel du domaine de recherche qui nous intéresse, à savoir les systèmes multi-agents. De plus, nous explorons plus en détail les deux types d'agents (cognitifs et réactifs). Dans la deuxième partie, nous allons justifier l'intérêt de la présence des comportements sociaux complexes dans les systèmes multi-agents. La troisième partie s'intéresse au problème de l'organisation dans les « systèmes ouverts ». Dans ce cadre, elle comprend l'étude et l'analyse des comportements sociaux complexes chez les primates qui peuvent émerger dans un système multi-agents. En effet, nous avons trouvé chez les primates des comportements sociaux correspondant assez bien aux besoins de l'intelligence artificielle distribuée en milieu ouvert. Cette complexification des agents enrichit leur comportement et donc les possibilités d'interaction. Cela donne accès à certaines caractéristiques sociales, telles que le regroupement et la spécialisation, le partage des ressources, etc.

4.2. Intelligence artificielle distribuée et systèmes multi-agents

Le cadre de recherche, dans lequel s'inscrit ce chapitre, est celui de l'intelligence artificielle distribuée, des systèmes multi-agents et des agents.

4.2.1. Intelligence artificielle distribuée

L'intelligence artificielle, discipline scientifique en pleine effervescence, a vu le jour au début des années 1970. À l'origine, elle s'est fixé comme objectif la résolution des

problèmes et des tâches complexes en simulant et en reproduisant le raisonnement d'un être humain. Après le succès des premiers systèmes experts, les applications de grande envergure, requérant à la fois une grande quantité de connaissances et des expertises multiples, ont très vite mis en évidence les limites de l'intelligence artificielle classique. Pour pallier ces manques, les chercheurs ont proposé de passer de la métaphore du penseur unique à celle d'une société d'experts (Russell et Norvig, 2004).

Cette nouvelle façon de penser constitue le fondement d'un nouveau paradigme, celui d'une société d'experts humains permettant de faire cohabiter des expertises multiples. On parle alors de systèmes de l'intelligence artificielle distribuée. L'intelligence artificielle distribuée permet donc de surmonter les difficultés d'un système intelligent unique (dans des domaines d'application complexes et hétérogènes tels que la reconnaissance des formes, l'aide à la décision, etc.) en distribuant l'expertise au sein d'une société de systèmes intelligents, au même titre que les êtres humains accomplissant collectivement des tâches complexes qu'ils ne peuvent accomplir individuellement. L'intelligence, dans ce cas, est vue comme la capacité d'un groupe d'experts à collaborer et à coordonner leurs actions pour arriver à accomplir leurs objectifs, aussi bien individuels que collectifs.

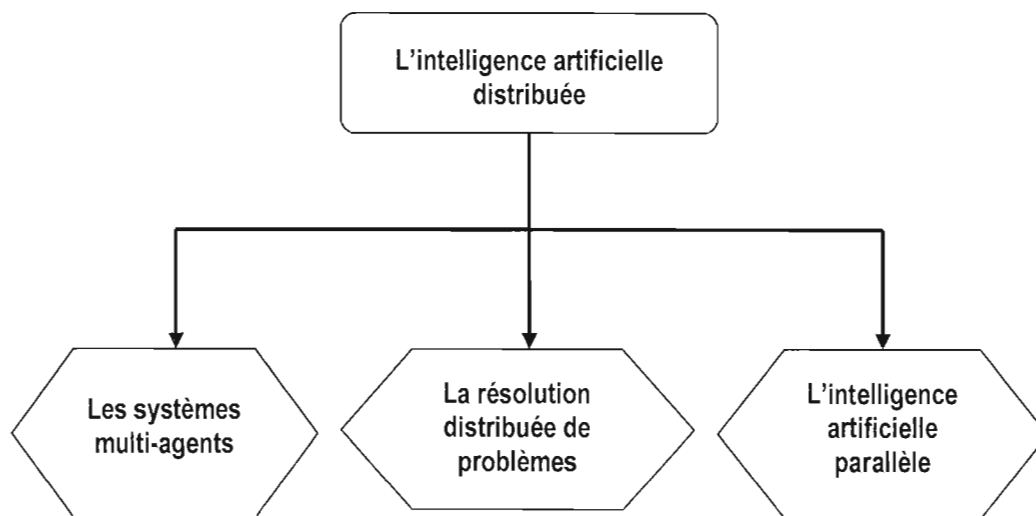


Figure 4.1 Les trois axes fondamentaux de recherche de l'intelligence artificielle distribuée

En résumé, l'intelligence artificielle distribuée est une branche de l'intelligence artificielle qui s'intéresse à la modélisation de comportements intelligents par la coopération d'un ensemble d'entités autonomes (ou semi-autonomes), qui interagissent entre elles et avec leur environnement. En réalité, l'intelligence artificielle distribuée, illustrée par la figure 4.1, recouvre trois axes fondamentaux de recherche (Russell et Norvig, 2004):

- ✓ Système multi-agents : il est un ensemble d'entités, appelées agents, dotées d'un comportement intelligent et interagissant, dans un environnement, selon une organisation pour atteindre un but commun.
- ✓ Résolution distribuée des problèmes : elle s'intéresse à la résolution coopérative par un traitement, distribué et flexible, des connaissances localisées au niveau des différentes entités. Ainsi, les entités, en opposition aux systèmes multi-agents, doivent coopérer pour résoudre un problème, parce qu'aucune ne possède la totalité de l'information.
- ✓ Intelligence artificielle parallèle : elle vise la parallélisation des approches existantes en intelligence artificielle. Pour ce faire, elle développe des langages et des algorithmes parallèles qui visent à améliorer les performances d'un système d'intelligence artificielle. Par opposition aux systèmes multi-agents, elle ne s'intéresse pas à la nature du raisonnement et au comportement intelligent d'un groupe d'agents.

4.2.2. Agents et systèmes multi-agents

Dans le reste de ce chapitre, nous allons nous intéresser uniquement aux systèmes multi-agents. Nous pouvons décrire ces derniers comme étant un ensemble d'entités actives et autonomes, les agents, qui évoluent dans un environnement. Selon Russell et Norvig (2004, p. 33), un agent, dans le cadre d'un système multi-agents, peut être défini comme suit :

« An agent is anything that can be viewed as perceiving its environment through sensors and acting upon that environment through effectors ».

En d'autres termes, un agent, illustré par la figure 4.2, est une entité qui perçoit grâce à ses capteurs et agit grâce à ces effecteurs dans un environnement pour accomplir ses buts.

L'ensemble des actions réalisées par un agent dans son environnement constitue son comportement. Ce dernier est généralement modélisé par le biais de l'intelligence artificielle classique. Ainsi, un agent accomplit des tâches d'une manière autonome et en fonction du but à atteindre. Pour ce faire, l'agent est doté de la capacité de se coordonner, collaborer, échanger des informations, etc. Dans le cadre d'un système multi-agents, l'agent, avant tout, a la vocation de composer, avec d'autres, une communauté d'agents autonomes travaillant en commun, selon des modes parfois complexes de coopération, conflit, concurrence, pour aboutir à un objectif global.

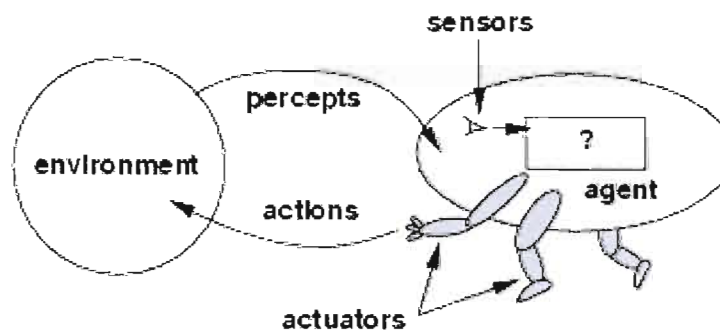


Figure 4.2 Agent d'après (Russell et Norvig, 2004)

Compte tenu de ce qui précède, une définition d'agent plus adéquate, introduisant le concept de systèmes multi-agents, s'impose. Ainsi, selon Ferber (2007), un agent est une entité réelle ou abstraite qui est capable d'agir sur elle-même et son environnement. Il dispose d'une représentation globale ou partielle de cet environnement ; qui peut, dans un univers multi-agents, communiquer avec d'autres agents, et dont le comportement est la conséquence de ses observations, de sa connaissance et des interactions avec les autres agents (Ferber, 2007).

Comme nous l'avons mentionné plus haut, pour que cette organisation se mette en place, il n'est pas toujours nécessaire que les agents disposent d'une vision globale de leur environnement. En effet, des études sur les systèmes multi-agents ont montré comment des

agents au comportement simple, ne possédant qu'une capacité locale de perception et d'action, sont capables de créer des structures organisationnelles complexes telles que la simulation de production de nids de guêpes (Bonabeau et Theraulaz, 1994) ou des termitières (Resnick, 1997). Ces agents dits réactifs tentent d'obtenir des comportements collectifs de haut niveau grâce à des interactions d'agents les plus simples possible. Dans *MANTA* (Drogoul et Dubreuil, 1992), la simulation d'une fourmilière en est une excellente illustration, dans la mesure où elle permet, entre autres, d'expliquer et de reproduire les répartitions des tâches et la spécialisation spatiale des fourmis en dépit de principes de parcimonie à l'échelle individuelle (Picault, 1998 ; Picault et Collinot, 1998).

4.2.3. Formes de l'intelligence artificielle distribuée

La littérature montre que dans l'univers de l'intelligence artificielle distribuée, nous pouvons distinguer deux courants extrêmes (Gechter et Simonin, 2005). D'une part, l'intelligence artificielle distribuée cognitive (ou sociale) est héritière en grande partie de l'intelligence artificielle classique. Elle s'intéresse à des architectures où chaque agent est assimilable, en fonction de ses capacités, à un système expert plus ou moins sophistiqué. En effet, les agents utilisent des bases de connaissances symboliques reposant sur des logiques modales, des mécanismes inférentiels, de la planification, des protocoles explicites de négociation et de coordination entre les agents, etc. (Jennings, 1993 ; Rao et Georgeff, 1991). En d'autres termes, les systèmes cognitifs comprennent un nombre de connaissances, une aptitude à traiter des informations diverses liées au domaine d'application, d'informations relatives à la gestion des interactions avec les autres agents et l'environnement. De ce fait, l'architecture d'un agent cognitif se distingue essentiellement par les connaissances de soi et des autres, l'expertise du domaine, les connaissances de communication et de coordination.

Pour conclure, selon Demazeau et Müller (1990), un agent cognitif doit, dans le cadre d'un système multi-agent, présenter les caractéristiques suivantes :

- ✓ Intentionnalité : d'après Searle (1969) :

*« Une intention est la déclaration explicite des buts et des moyens d'y parvenir.
Elle exprime la volonté d'un agent d'atteindre un but ou d'effectuer une action »*

Plus précisément, l'intention d'un agent est la déclaration explicite des objectifs individuels. Ainsi, elle exprime la volonté consciente d'un agent d'atteindre un but ou d'effectuer une tâche.

- ✓ Rationalité : selon Newell (1982), le principe de rationalité consiste à faire en sorte que :

« Si un agent sait que l'une de ses actions lui permet d'atteindre ses buts, alors il sélectionnera cette action ».

En d'autres termes, un agent rationnel dispose de moyens d'évaluation de ses actions et ainsi, peut décider de la plus pertinente. Donc, les actions sont toujours le fruit d'un raisonnement et servent à satisfaire les buts de l'agent (Ferber, 2007).

- ✓ Engagement : la notion d'engagement est l'une des qualités essentielles des agents coopératifs. Un agent coopératif planifie ses actions par coordination et négociation avec les autres agents (Ferber, 2007).
- ✓ Adaptation : un agent adaptatif est un agent capable de contrôler, modifier et ajuster ses aptitudes communicationnelles et comportementales en fonction des autres agents et de l'environnement. Un agent adaptatif est un agent de haut niveau de flexibilité.
- ✓ Intelligence : Un agent intelligent est un agent capable d'action autonome flexible dans un certain environnement. Par flexible, on entend réactif, proactif et social. Selon Demazeau et Müller (1990), un agent intelligent est un agent cognitif, rationnel, intentionnel et adaptatif.

D'autre part, l'école réactive (ou biologique) présuppose qu'il n'est pas nécessaire à chaque agent d'être individuellement « intelligent » pour parvenir à un comportement global intelligent (Demazeau et Müller, 1991 ; Ferber, 2007). Dans ce but, des mécanismes de réaction aux événements peuvent faire émerger, par effet secondaire, des comportements correspondants à l'objectif poursuivi. De plus, l'explication de ces mécanismes ne peut se

faire ni en termes de buts, ni de mécanismes de planification. Par conséquent, un agent réactif est donc un agent très simple ne possédant pas de représentation explicite de son environnement, et dont le comportement est primitif et ne consiste en général qu'à répondre selon la loi stimulus/action. On parle alors d'agents à faible granularité (faible complexité). En d'autres termes, ils ne disposent que d'un langage et d'un protocole de communication réduits. Ainsi, l'approche réactive utilise des agents ne disposant que des capacités de perception et d'actions locales, sans aucune connaissance explicite sur leur environnement.

D'un point de vue structurel, un agent réactif peut être modélisé comme un simple objet doté d'un comportement et d'un moyen de communication avec les autres agents. Il n'a ni connaissances sur les autres, ni capacité de raisonner sur les messages qu'il reçoit, ni développement de stratégies de contrôle (Demazeau et Müller, 1991). Pour conclure, contrairement à l'approche cognitive définie plus haut (système cognitif), un agent réactif est plutôt centré sur le comportement que sur la connaissance. En effet, les connaissances sur les relations s'expriment plus par les interactions entre les agents que, par une représentation explicite de ces dernières. Bien que les agents réactifs soient faciles à mettre en œuvre, l'école cognitive reste la plus utilisée.

Dans un même ordre d'idées, il existe une deuxième différence, basée quant à elle sur le champ d'application des systèmes considérés. En effet, les agents cognitifs doivent répondre à des contraintes précises préalablement établies afin d'obtenir une ou des fonctionnalités spécifiques. Ainsi, ils ne peuvent être employés que dans un cadre assez délimité, prévisible, peu dynamique. En d'autres termes, ils ne peuvent être utilisés que dans un « environnement clos ». Par contre, les agents réactifs échappent, généralement, à ces contraintes. De cette façon, ils bénéficient d'une marge de manœuvre plus grande dans leur champ d'application. En d'autres termes, ils peuvent être utilisés dans un environnement moins limité : un « environnement ouvert ». En effet, dans un environnement ouvert, aucun organe de régulation n'est présent dans le système. Le tableau 4.1 ci-dessous résume les différences entre les agents cognitifs et réactifs.

Tableau 4.1 Les différences entre les agents cognitifs et réactifs

<i>AGENTS COGNITIFS</i>	<i>AGENTS RÉACTIFS</i>
Représentation explicite de l'environnement	Pas de représentation explicite de l'environnement
Peut tenir compte de son passé	Pas de mémoire de son historique
Agents complexes	Fonctionnement stimulus/réponse
Petit nombre d'agents de forte granularité	Grand nombre d'agents de faible granularité
Applicable au sein d'un « environnement clos »	Applicable au sein d'un « environnement ouvert »

4.3. Intérêt des comportements sociaux complexes dans les systèmes multi-agents

L'un des problèmes cruciaux des systèmes multi-agents est celui de l'organisation des agents. Cette dernière, en dépit de son importance, reste jusqu'à présent une notion mal définie. L'organisation au sein d'un système multi-agents implique que ces systèmes réussissent à se coordonner, collaborer, gérer l'accès aux ressources, résoudre les conflits, échanger des informations en fonction du but à atteindre. À cet effet, des contraintes sont, généralement, incorporées aux comportements individuels des agents. Ces contraintes sont composées de règles explicites sur les interactions entre agents. Ainsi, nous pouvons limiter leur marge d'action dans le but d'obtenir une certaine forme de coordination, donnant lieu à un comportement collectif intéressant.

En fait, l'organisation est ce qui distingue un système multi-agents d'un système parallèle. Dans ce dernier, comme dans les systèmes d'exploitation, les conflits, à la suite de l'accès aux ressources partagées par un ensemble de processus, sont généralement gérés par un mécanisme centralisé de synchronisation. En revanche, dans un système multi-agents, les conflits entre agents sont gérés d'une manière non centralisée. En effet, les agents sont considérés comme des entités autonomes. De plus, l'organisation intervient aussi bien dans la coordination et la coopération entre agents pour accomplir une tâche commune que dans la prévention et la résolution de conflits.

Pour conclure, l'organisation au sein d'un groupe agent peut être considérée comme une organisation sociale, comme dans le cadre d'une communauté humaine ou animale. En d'autres termes, cette dernière est un ensemble de contraintes auxquelles sont assujetties l'ensemble du groupe, régissant les interactions entre ces membres. Par conséquent, les comportements auxquels les agents ont recours pour instaurer l'organisation au sein du système sont assimilables à des « comportements sociaux ».

Généralement, l'organisation au sein d'un système multi-agents est abordée au moyen des métaphores sociales correspondant à l'un des deux paradigmes abordés dans la section précédente. En effet, les règles peuvent 1) être fixées explicitement, avec une valeur normative (école cognitive), en tant qu'ensemble de connaissances et de contraintes précises qui sont préalablement établies et qui vont fournir un cadre aux actions des agents, ou 2) au contraire être inscrites implicitement dans les comportements individuels (école réactive) et dans les modes de communication entre agents (Quinqueton et Hamadi, 1999). Dans le second cas, on parle en général d'organisation émergente (Visetti, 2004), au sens où le comportement collectif qui peut être observé n'est pas immédiatement déductible de l'ensemble des comportements individuels.



Figure 4.3 Pointillisme

La notion d'émergence est présente, dans la littérature, depuis longtemps. Nous pouvons comparer ce concept au pointillisme en peinture. Cette technique, illustrée par la figure 4.3, consiste à peindre sur une toile une multitude de petits points en se servant uniquement de couleurs primaires. Ainsi, lorsqu'il observe l'œuvre à une certaine distance, l'œil perçoit des tons secondaires, grâce à une illusion d'optique causée par la juxtaposition des points colorés.

Une des premières définitions de l'émergence a été donnée par Minsky (1986), qui la définit comme suit :

« Apparition inattendue, à partir d'un système complexe, d'un phénomène qui n'avait pas semblé inhérent aux différentes parties de ce système. Ces phénomènes émergents ou collectifs montrent qu'un tout peut être supérieur à la somme de ses parties. »

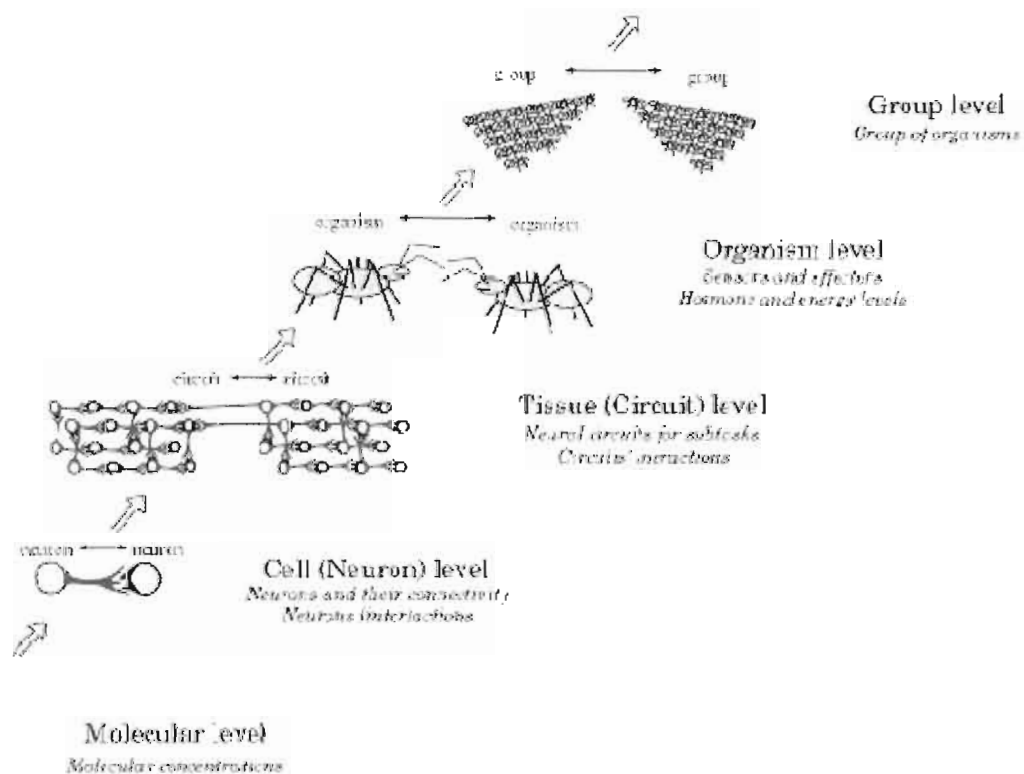


Figure 4.4 Les niveaux hiérarchiques des organismes d'après Vaario (1994)

Cette définition est intéressante, mais présente une lacune. Elle définit l'émergence comme une apparition inattendue. Or, elle n'a pas besoin d'être inattendue pour être de l'émergence. De ce fait, nous pouvons redéfinir le concept d'émergence, donné par la figure 4.4, comme l'existence dans un système des propriétés qui sont présentes à un certain niveau d'organisation et qui ne peuvent être déduites des propriétés de niveaux inférieurs. Si nous reprenons le parallèle avec le pointillisme, nous ne pouvons pas déduire la peinture à partir de la propriété des éléments du niveau inférieur, c'est-à-dire les points.

Dans le domaine des systèmes multi-agents, la définition la plus utilisée est la généralisation du calcul émergent proposée par Müller (1998). Un phénomène est dit émergent si :

- ✓ Il y a un ensemble d'agents en interaction entre eux et avec l'environnement dont la dynamique n'est pas exprimée dans les termes du phénomène émergent à produire, mais dans un vocabulaire ou une théorie « D » ;
- ✓ La dynamique des agents en interaction produit un phénomène global qui peut être une structure stable, une trace d'exécution ou n'importe quel invariant statique ou dynamique ;
- ✓ Ce phénomène global est observé soit par un observateur extérieur, soit par les agents eux-mêmes en des termes distincts de la dynamique sous-jacente, c'est-à-dire avec un autre vocabulaire ou théorie « D' ».

Dans le domaine des systèmes multi-agents, de nombreux travaux ont montré que des organisations pouvaient émerger des interactions entre agents. Par exemple, dans la simulation de robots explorateurs (Drogoul et Ferber, 1993), les phénomènes de pistes ou de chaînes émergent des coordinations locales qui ont lieu entre les individus. Dans le même ordre d'idées, dans la simulation des comportements au sein d'une fourmilière (Drogoul et Ferber, 1994), les interactions individuelles donnent naissance à des phénomènes d'organisation du travail et de structuration des rôles. Enfin, dans les expériences « Talking heads » (Steels, 2000, 2003b, 2003a), les interactions individuelles ont permis la formation automatique d'un vocabulaire, et l'apprentissage individuel et en groupe du langage.

Généralement, dans les systèmes multi-agents, nous nous limitons aux phénomènes qui émergent des interactions individuelles au niveau des trois aspects de coopération suivants :

- ✓ Le regroupement : il consiste à rapprocher les agents, soit physiquement, soit via un réseau de communication.
- ✓ La spécialisation : elle restreint temporairement les agents à des rôles particuliers par une sorte d'adaptation individuelle. Cette spécialisation peut être bénéfique à la collectivité en augmentant la capacité du groupe à résoudre plus rapidement un problème.
- ✓ Le partage des tâches et des ressources : il permet à des agents de se répartir des tâches, des informations et des ressources de manière à réaliser un objectif commun.

Toutefois, d'un point de vue pratique, la première solution envisagée est héritée de l'intelligence artificielle symbolique et consiste dans des modèles d'agents cognitifs disposant d'une intentionnalité et de mécanismes de planification plus ou moins sophistiqués. La coopération entre congénères est alors mise en œuvre au moyen de protocoles d'allocation de tâches, de négociation, de synchronisation, etc. (Ferber, 2007). Le paradigme sous-jacent est celui de la communauté d'experts. Les organisations normatives ne peuvent être employées que dans un cadre bien spécifique, prévisible, peu dynamique, car en permettant l'introduction de règles explicites sur les interactions entre agents, elles facilitent la conception du système. Cependant, rien n'assure que la forme imposée aux relations sociales soit adaptée fonctionnellement aux événements pouvant survenir ultérieurement dans l'environnement (Agre et Chapman, 1995).

En revanche, l'intelligence artificielle distribuée réactive adopte un autre point de vue : il n'est pas nécessaire d'utiliser des entités individuellement complexes pour obtenir un résultat collectif intelligent. La simulation d'une fourmilière, dans « Manta » (Drogoul, 1993), en est une excellente illustration dans la mesure où elle permet d'expliquer et de reproduire la répartition des tâches et la spécialisation spatiale des fourmis. Cette approche donne lieu non seulement à la modélisation de phénomènes biologiques ou sociologiques, mais aussi à de

nombreuses applications pour la résolution de problèmes, la robotique collective, les stratégies d'évitement d'obstacles, etc.

Cependant, en dépit des efforts méthodologiques amorcés dans ce sens, les organisations émergentes, si elles permettent de représenter par exemple le fonctionnement de systèmes complexes existants dans la nature, posent de sérieux problèmes. Ces problèmes résident essentiellement dans la conception d'un système multi-agent en vue d'une application spécifique (Gechter et Simonin, 2005 ; Müller, 1998). Cela dit, si l'on est capable de donner au système le moyen de s'organiser lui-même, la forme de cette organisation sera, en théorie, adaptée à l'application. Ainsi, l'organisation devrait refléter les contraintes imposées aux comportements des agents par l'application (Bonabeau et Theraulaz, 1994).

4.4. Phénomènes sociaux

Nous nous intéressons principalement à la conception de systèmes complexes dont le comportement global émerge du comportement des agents qui les composent et de leurs interactions. Une telle conception doit enrichir le champ d'action de l'agent dans son environnement et favoriser l'émergence de comportements sociaux complexes (phénomènes et organisations sociaux). Il réagira ainsi plus adéquatement aux événements qui surviendront dans son environnement.

Le but de cette section est d'identifier quelques comportements sociaux chez les primates susceptibles d'être produits par un système multi-agents. Le choix des primates, plutôt que d'autres animaux (loups, fourmis, araignées, dauphins, etc.), se justifie par le fait que les comportements sociaux chez les primates correspondent assez bien aux besoins de l'intelligence artificielle distribuée en milieu ouvert. Les interactions chez les primates, d'une complexité inégalée (une large gamme de comportements sociaux), se rapprochent à des agents opérant dans un milieu ouvert et socialisé. Les compétences sociales des primates non humains sont en effet très proches de celles de l'homme et résolvent des problèmes organisationnels qu'on rencontre en intelligence artificielle distribuée (hiérarchie, reconnaissance de relation entre congénères, alliances, coopération, allocation de tâches, tromperie, etc.).

Une revue de la littérature, dans ce domaine, nous a permis d'identifier quelques phénomènes sociaux naturels capables de nous intéresser :

La représentation des relations existant entre les congénères. Chez les primates, les individus d'un groupe se connaissent mutuellement ainsi que les relations sociales existantes (dominance, parenté, affiliation). De plus, les individus peuvent tenir compte dans leurs comportements de la transitivité de la hiérarchie (si *A* domine *B* et *B* domine *C*, alors généralement *A* domine *C*). Ceci implique, selon Dasser (1988), que les primates disposent d'une représentation symbolique des relations sociales et d'un mécanisme d'inférence.

Des comportements d'agrégation. Les comportements les plus courants chez les primates sont les comportements « affiliatifs » (épouillage mutuel, jeu, etc.). L'objectif essentiel de ces derniers est le renforcement des liens sociaux. Il est à noter que les individus sont portés à demander leurs anciens partenaires (stratégie de type « donnant donnant ») pour ce genre de comportement (De Waal, 2000). La figure 4.5 représente un comportement « affiliatif » : l'épouillage mutuel.

Des comportements de coopération. Chez les primates, les individus d'un groupe sont capables de former des coalitions pendant les combats, et d'utiliser des réseaux d'alliances pour le maintien (macaques, babouins) ou le renversement (chimpanzés) de l'ordre établi. Ces réseaux se constituent et évoluent en fonction des liens de parenté, mais aussi des interactions affiliatives entre individus (Drogoul et Collinot, 1997 ; Kappeler, 2005). En fait, nous retrouvons ce type de comportement chez d'autres types d'animaux, comme les dauphins (Connor, 2007 ; Connor, Smolker et Bejder, 2006). L'exemple le plus connu est l'expérience, illustrée par la figure 4.6, où deux chimpanzés doivent travailler ensemble pour obtenir une récompense de nourriture. Ni l'un ni l'autre n'a la force de tirer la corde pour déplacer la boîte chargée nourriture, mais en travaillant ensemble (coopération), ils peuvent le faire.



Figure 4.5 Un exemple de comportements d'agrégation (De Waal, 2000)

Des comportements de coordination. Les techniques de chasse en groupe chez les chimpanzés, décrites par Boesch (1989), en sont l'illustration parfaite. En effet, les chasseurs encerclent leurs proies et anticipent leurs déplacements. Dans ce but, la coordination doit être gérée de manière parfaite entre les participants. Nous retrouvons ce type de comportement chez d'autres types d'animaux, comme les dauphins (Connor, 2007 ; Connor, Smolker et Bejder, 2006).

L'allocation dynamique de tâches en fonction des besoins du groupe et des compétences individuelles. La chasse en fournit également un exemple. En effet, les chimpanzés doivent jouer un certain nombre de rôles bien définis (rabatteur, poursuivant, etc.), qu'ils doivent se répartir dynamiquement en fonction du nombre de chasseurs, de l'évolution de la chasse, et des compétences de chacun (Drogoul et Collinot, 1997). Comme les comportements de coopération et de coordination, nous retrouvons ce type de comportement chez d'autres types d'animaux, comme les dauphins (Connor, 2007 ; Connor, Smolker et Bejder, 2006).



Figure 4.6 Un exemple de comportements de coopération (Kappeler, 2005)

La réduction des tensions sociales. De Waal (2000) explique comment les tensions sociales engendrées par les conflits et la hiérarchie sont éliminées. Dans un système multi-agents capable d'utiliser dans son fonctionnement des conflits entre agents, il est nécessaire d'éviter que les tensions ne s'accumulent de manière excessive, sous peine de devenir le problème central des agents, au détriment de leur tâche collective initiale (Galliers, 1990).

La tromperie sociale. La tromperie sociale chez certaines espèces de primates est largement documentée (Byrne, 1995, 2006 ; Byrne et Bates, 2007 ; Byrne et Whiten, 1988 ; Castelfranchi, 2000). Elle consiste à manipuler les congénères à travers l'utilisation « hors contexte » de comportements habituels. Il s'agit en fait d'une manipulation liée à la connaissance de l'effet que peuvent avoir les comportements d'un agent sur ses congénères. En effet, parfois une gorille femelle pourrait rencontrer un mâle en cachette (par exemple

derrière un rocher, illustré par la figure 4.7) pour éviter qu'elle soit battue par le mâle dominant. De plus, un singe pourrait même feindre son désintéressement à de la nourriture savoureuse pour éviter que d'autres ne viennent lui voler (Byrne, 1995, 2006 ; Byrne et Bates, 2007 ; Byrne et Whiten, 1988 ; Castelfranchi, 2000).

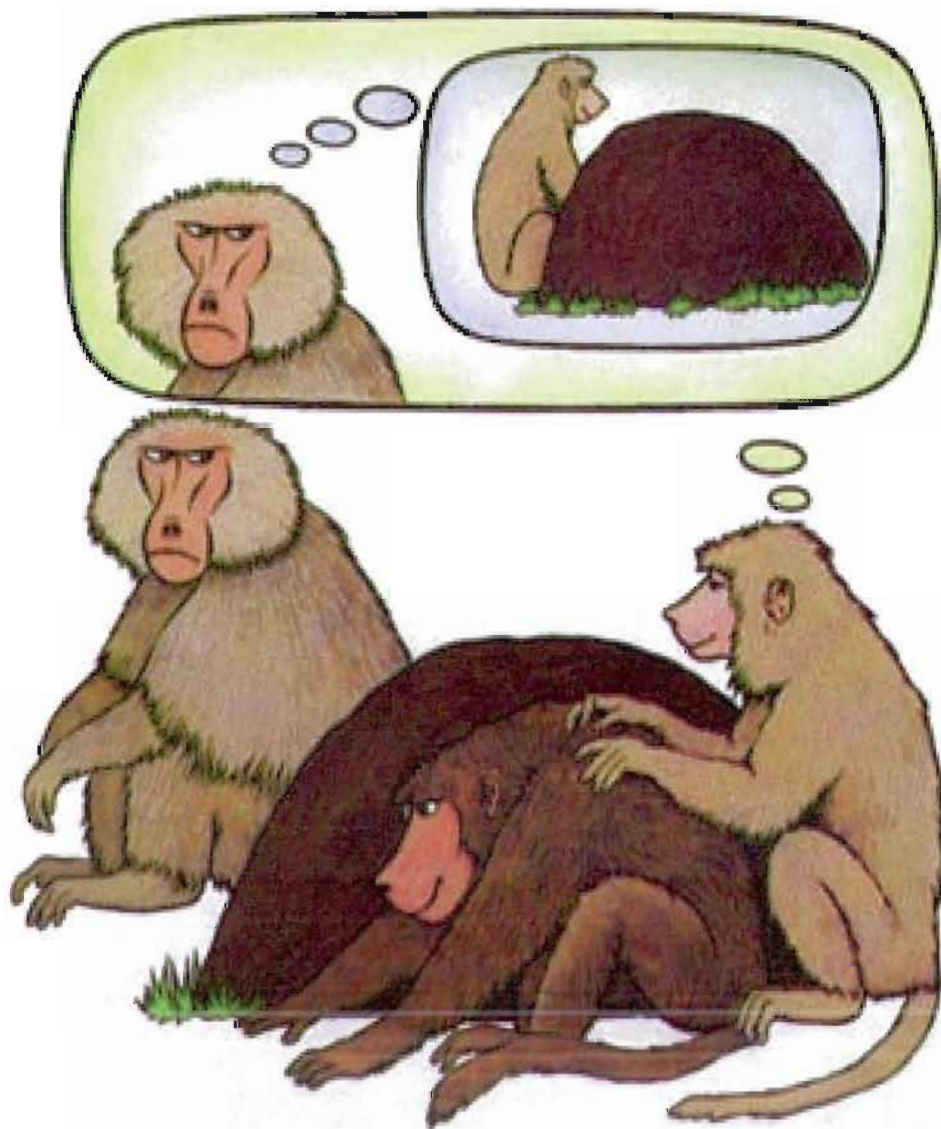


Figure 4.7 Un première exemple de tromperie sociale d'après (Byrne, 1995)

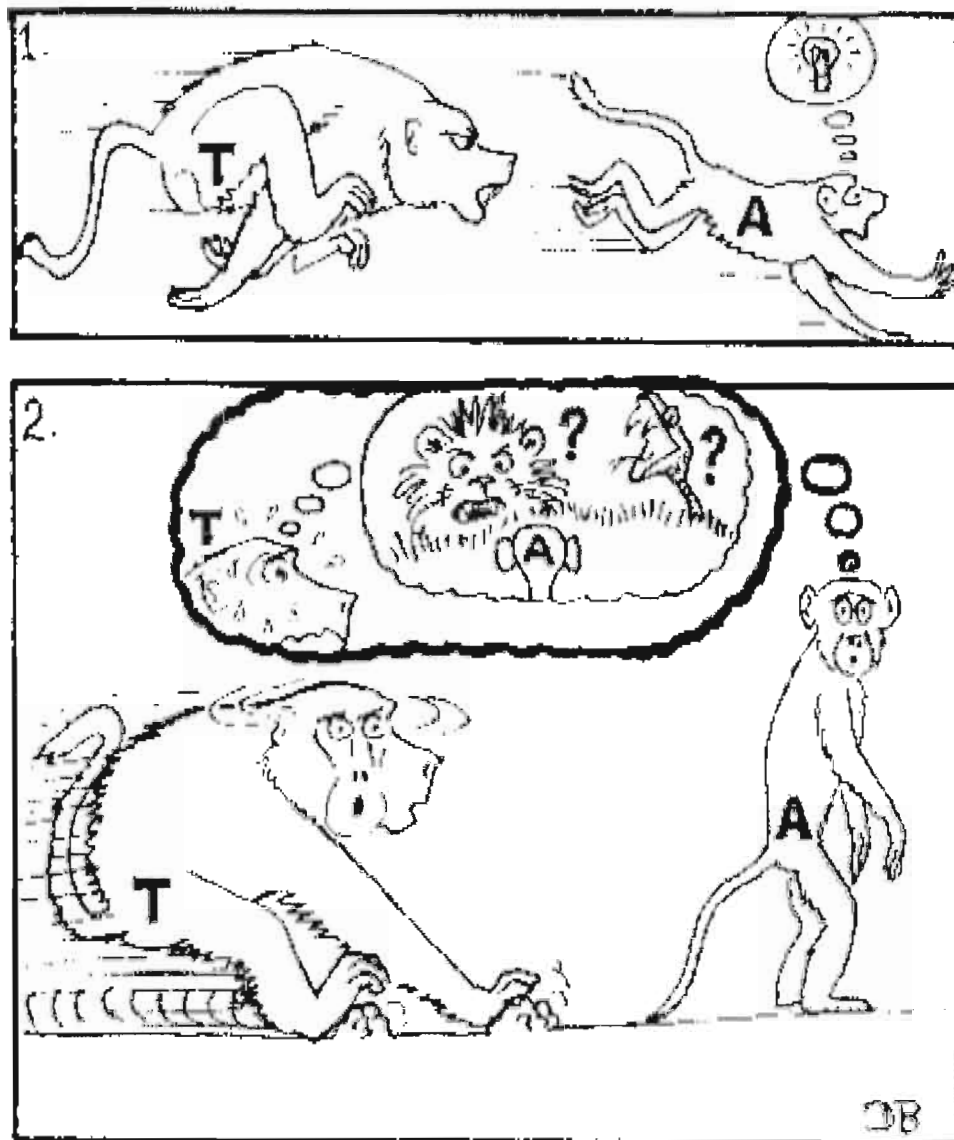


Figure 4.8 Un deuxième exemple de tromperie sociale d'après (Byrne, 1995)

La figure 4.8 présente un exemple de tromperie sociale. En effet, un mâle *A* est poursuivi par un adulte *T*. Mais au lieu de continuer à courir, *A* se dresse soudain sur ses pattes postérieures et fixe l'horizon, comme le fait tout mâle qui perçoit un danger. Son poursuivant s'arrête pensant qu'un danger les guettait et regarde dans la même direction. Bien qu'il n'y ait rien de suspect, la poursuite s'arrête (Byrne, 1995).

La figure 4.9 présente un autre exemple de tromperie sociale. En effet, un mâle *A* a envie de s'emparer de la nourriture d'un adulte *T*. *A* commence soudain à crier, comme le fait tout mâle agressé par un adulte. L'adulte *TOOL* suppose que le mâle *A* a été agressé par l'adulte *T* et poursuit ce dernier. Ainsi, le mâle *A* a accès à la nourriture (Byrne, 1995).

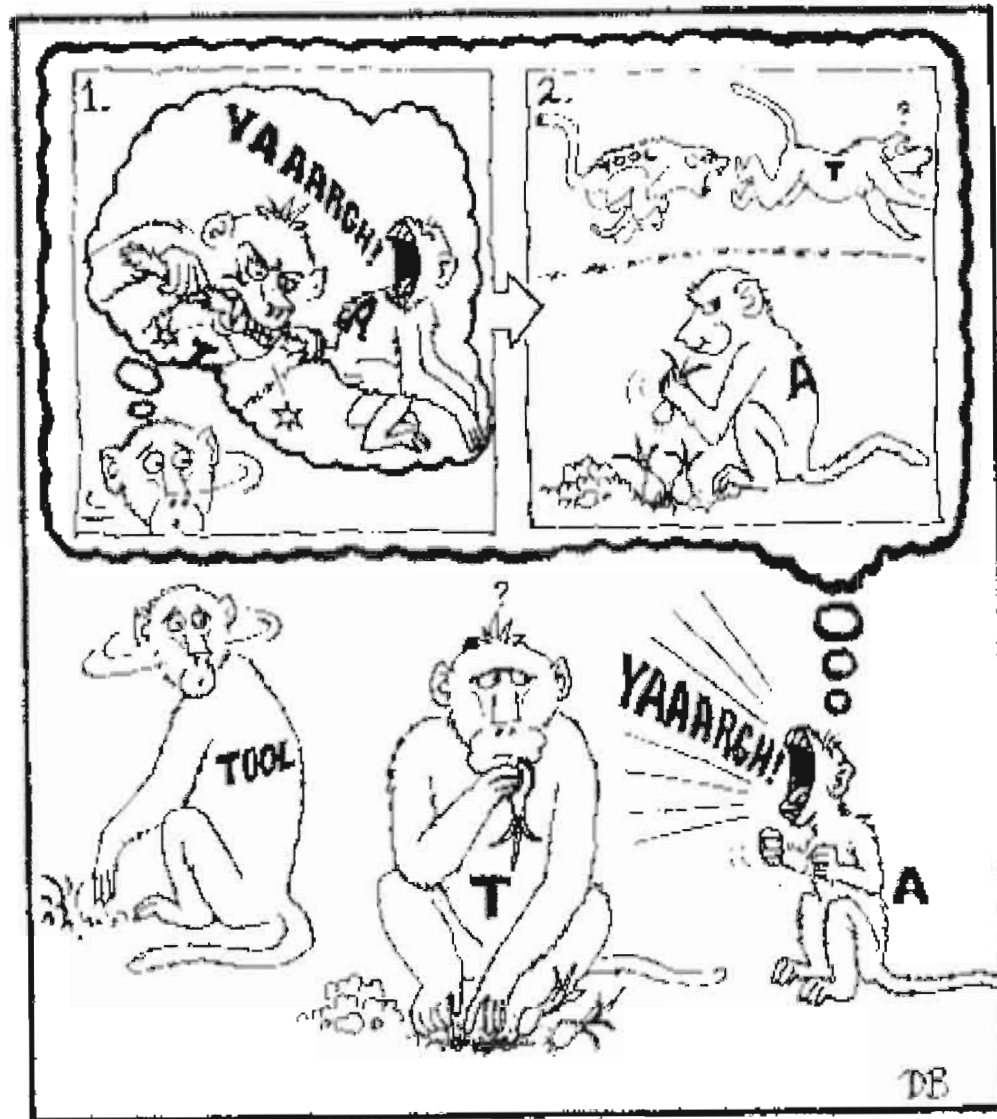


Figure 4.9 Un troisième exemple de tromperie sociale d'après (Byrne, 1995)

4.5. Conclusion

Le travail présenté dans ce chapitre s'intéresse au problème de l'organisation dans les « systèmes ouverts ». Dans ce cadre, cette recherche vise à définir des modèles de comportements sociaux permettant aux agents de s'organiser pour s'adapter à leur environnement. Notre démarche fait appel, entre autres, au concept qui se propose d'emprunter des métaphores à l'éthologie pour concevoir des modèles informatiques. Pour ce faire, ce chapitre nous a conduit, dans un premier temps, à introduire les concepts d'agent et de systèmes multi-agents. Ensuite, nous avons pu nous intéresser à quelques mises en œuvre de simulation multi-agents de modèles de cognition sociale, et ceci, dans les deux courants à savoir : l'école cognitive et l'école réactive. Enfin, nous avons pu identifier des comportements sociaux intéressants chez les primates susceptibles d'émerger au sein d'un système multi-agents.

Cependant, la majorité des études portant sur ce sujet de recherche s'intéresse essentiellement à des implémentations mettant en jeu des techniques issues de l'intelligence artificielle classique. Cela pose évidemment, d'une part, le problème de l'ancrage des représentations et, d'autre part, le problème de l'utilisation de ces systèmes dans des environnements autres que clos. En effet, les approches de l'école cognitive ne peuvent être utilisées que dans un cadre bien spécifique, prévisible, peu dynamique. En effet, en permettant l'introduction de règles explicites sur les interactions entre agents, l'école cognitive facilite sans doute la conception du système, mais rien n'assure que ces règles soient adaptées fonctionnellement aux événements pouvant survenir ultérieurement dans l'environnement. Néanmoins, nous avons pu voir que les approches alternatives, inspiré de l'école réactive, ont fourni des résultats très intéressants dans ce domaine de recherche. C'est dans cette optique que nous pensons que les systèmes de recommandation devraient tenir compte de l'aspect comportements sociaux. En effet, nous pensons qu'un tel système pourrait donner des résultats intéressants. Cependant, il est à noter que, faute de temps, l'approche de système de recommandation présentée dans le chapitre suivant ne prend pas en considération cet aspect comportements sociaux. Nous essayerons de traiter ce dernier dans nos futures recherches.

Pour conclure, il est à noter que l'étude des comportements sociaux ne doit pas se limiter qu'aux primates. En effet, l'organisation chez d'autres animaux (loups, fourmis, araignées, dauphins, etc.) nous a montré qu'ils sont capables de créer des structures sociales complexes (Bonabeau et Theraulaz, 1994 ; Drogoul, 1993 ; Resnick, 1997).

Deuxième partie

Application

Chapitre V

Proposition de recherche

« Une proposition incorrecte est forcément fausse, mais une proposition correcte n'est pas forcément vraie. »

– Emmanuel Kant (1724 – 1804)

5.1. Introduction

Compte tenu des chapitres précédents, nous nous sommes proposé d'approfondir quelques points de recherche. En premier lieu, nous avons décidé de porter notre attention sur les systèmes de filtrage de l'information hybride. En outre, nous allons nous pencher sur l'incorporation d'une architecture pair-à-pair dans notre système de recommandation. Enfin, nous allons explorer l'adoption de comportements sociaux complexes dans notre système multi-agent. À cet effet, le modèle cognitif que nous proposons est une plate-forme d'agents collaboratifs pour le filtrage de l'information hybride basée sur une architecture pair-à-pair. Les points clés de ce modèle sont :

Filtrage hybride. Nous proposons une méthode hybride qui allie le meilleur des deux approches classiques (filtrage collaboratif et filtrage basé sur le contenu). Nous évitons ainsi les problèmes de masse critique et de démarrage à froid. L'originalité de notre approche vient de l'utilisation d'une technique de discrimination du voisinage. Notre hypothèse est que cette technique nous permettra, d'une part, de réduire la complexité computationnelle du système de recommandation et d'autre part, améliorer la qualité de la recommandation.

Architecture distribuée. Nous avons opté pour une architecture pair-à-pair décentralisée. Cette architecture complètement distribuée est dépourvue d'autorité centrale (serveur

central). En effet, chaque pair (nœud) agit comme un serveur et un client. Dans notre approche, chaque utilisateur représente un pair distant du réseau. Le choix d'une telle architecture se justifie par le fait de contrer les contraintes de coûts, de robustesse, et de sécurité.

Système multi-agent. Notre système de recommandation est conçu comme une plateforme d'agents intelligents collaboratifs. En effet, nous allons considérer le système comme un ensemble d'agents réactifs indépendants, un agent pour chaque utilisateur du système. L'hypothèse derrière ce choix est qu'un tel système pourrait être capable de s'auto-organiser en fonction de l'environnement et ainsi, générer des comportements sociaux complexes.

Dans le reste de ce chapitre, nous allons voir les détails de mise en œuvre de notre approche. Plus précisément, nous allons présenter, dans un premier temps, la méthode choisie pour combiner le filtrage basé sur le contenu et le filtrage collaboratif. Ensuite, nous allons voir comment le choix d'une architecture décentralisée affectera la structure du système de filtrage. Enfin, nous allons introduire notre plateforme multi-agent.

5.2. Méthode de filtrage

Dans le chapitre 2, nous avons constaté que, dans le contexte de recommandation d'article, le filtrage collaboratif peut s'avérer très utile pour recommander aux utilisateurs certains articles qu'ils n'ont pas encore consultés, et dont on sait qu'ils intéressent leur groupe. En revanche, identifier un profil thématique pour chaque utilisateur permet d'évaluer si un texte est pertinent ou non pour cet utilisateur, en fonction de ses centres d'intérêt identifiés. Or, ces deux aspects du profil paraissent importants. Connaître les goûts thématiques personnels paraît la meilleure méthode pour fournir une aide spécialisée à l'utilisateur, mais associer les utilisateurs entre eux permet de faire bénéficier chacune des opinions des autres. Ainsi, nous envisageons donc d'utiliser le contenu des documents lus par les utilisateurs pour aider dans le filtrage collaboratif. Cette approche est appelée filtrage collaboratif via le contenu (Pazzani, 1999 ; Pazzani et Billsus, 1997).

Tableau 5.1 Filtrage collaboratif basé sur les notes

	Article 1	Article 2	Article 3	Article 4	Article 5
Utilisateur 1			7	6	???
Utilisateur 2			5	6	1
Utilisateur 3			6	6	3
Utilisateur 4	7	5			6
Utilisateur 5	7	6			4

Le filtrage collaboratif tel que nous l'avons présenté dans le chapitre 2 utilise l'ensemble des évaluations des utilisateurs sur les articles pour leur donner un degré de similarité. La matrice de données utilisée ressemble alors au tableau 5.1. Dans cet exemple, nous cherchons l'évaluation que l'utilisateur 1 va octroyer à l'article 5, en fonction des évaluations que les autres utilisateurs ont données à l'article, et de la similarité entre leurs vecteurs d'évaluations sur les autres articles.

Tableau 5.2 Filtrage collaboratif basé sur le contenu

	Mot 1	Mot 2	Mot 3	Mot 4	Mot 5	Article 5
Utilisateur 1	2.5	0	0.2	0	0	???
Utilisateur 2	1.1	0	1.1	1.5	0	1
Utilisateur 3	1.5	0	3.5	1.5	0.5	3
Utilisateur 4	1.1	1.1	2.1	2.0	2.5	6
Utilisateur 5	1.1	2.2	0	0	3.5	4

L'idée du filtrage collaboratif via le contenu est d'utiliser les profils thématiques des utilisateurs, formés à partir de l'analyse du contenu des articles qu'ils ont lus, pour comparer les utilisateurs dans la phase de calcul des corrélations entre utilisateurs du filtrage collaboratif (Pazzani, 1999 ; Pazzani et Billsus, 1997). La matrice de données utilisée ressemble alors au tableau 5.2. Dans cet exemple, nous cherchons l'évaluation que

l'utilisateur 1 va donner à l'article 5, en fonction des évaluations que les autres utilisateurs ont attribuées à l'article, et de la similarité entre leurs vecteurs de mots, représentant l'ensemble des articles lus. Le reste de cette section présente les paramètres de notre approche.

5.2.1. Méthode d'évaluation des documents

Nous allons adopter une approche hybride pour la collecte du degré d'appréciation des utilisateurs pour une ressource donnée. En effet, nous nous baserons, autant que possible, sur des évaluations explicites des utilisateurs. Le cas échéant, l'évaluation se fera sans la participation de l'utilisateur. Nous voulons éviter d'obliger l'utilisateur à participer dans la définition de son profil : comme nous l'avons vu, au lieu de lui demander si un article lui a plu ou non, on peut l'inférer, en se servant des informations que l'on peut obtenir lors de la consultation de l'article. Pour ce faire, nous pouvons utiliser le degré d'intérêt d'une page défini par Chan (1999) et introduit dans la section 2.3 :

$$\begin{aligned}
 Interest(Page) &= Frequency(Page) \\
 &\quad * (1 + IsBookmark(Page) + Duration(Page) + Recency(Page) \\
 &\quad + LinkVisitPercent(Page))
 \end{aligned}$$

Où

- ✓ $Frequency(Page)$ est la fréquence de visite de la page;
- ✓ $IsBookmark(Page) = \begin{cases} 1 & \text{si la page appartient au marque - pages} \\ 0 & \text{sinon} \end{cases}$
- ✓ $Duration(Page) = \frac{TotalDuration(Page) / Size(Page)}{\max_{Page \in VisitedPage} TotalDuration(Page) / Size(Page)}$
- ✓ $Recency(Page) = \frac{Time(LastVisit) - Time(StartLog)}{Time(Now) - Time(StartLog)}$
- ✓ $LinkVisitPercent(Page) = \frac{NumberOfLinksVisited(Page)}{NumberOfLinks(Page)}$

5.2.2. Représentation adoptée pour le profil thématique

La représentation adoptée pour le profil thématique est composée d'un profil initial rempli par l'utilisateur lui-même. Ce profil règle les problèmes de la masse critique et de démarrage à froid. En effet, nous allons opter pour une approche en profondeur qui indexera, après une phase de prétraitement du texte décrite dans le chapitre 2, l'intégralité du texte (« *bag-of-words* »). Ensuite, nous utiliserons un profil thématique à long terme, représenté sous la forme de vecteurs des mots présents dans l'ensemble des articles consultés par l'utilisateur. Enfin, nous aurons recours à un profil thématique à court terme, représenté sous la forme de vecteurs des mots présents dans les 10 derniers articles lus par l'utilisateur. Avec cette représentation, l'utilisateur ne serait donc pas limité à un profil fixe, mais l'ensemble des thèmes abordés serait toujours traité par le système de recommandation. Ainsi, les profils de l'utilisateur sont générés par le biais de l'algorithme de *Rocchio* en fonction du *TF-IDF* des termes (Adomavicius et Tuzhilin, 2005 ; Zaier, Godin et Faucher, 2007, 2008b):

$$w_j = \alpha * w_{1,j} + \beta * \frac{\sum_{i \in C} x_{i,j}}{n_c} - \gamma * \frac{\sum_{i \notin C} x_{i,j}}{n - n_c}$$

Où, pour un utilisateur donné j , C est l'ensemble des articles pertinents, n est le nombre total d'articles, n_c est le nombre d'articles pertinents, $w_{1,j}$ est le poids existant, et le $x_{i,j}$ est le *TF-IDF* de chaque terme de l'article i . Dans notre cas, nous avons pris les coefficients $\alpha=0$, $\beta=1$ et $\gamma=1$. Finalement, nous ajoutons au profil thématique d'utilisateur quelques informations démographiques normalisées sur les utilisateurs (âge, genre, métier, code postal, etc.). Il est à noter qu'étant donné que la pondération n'est pas le point central de cette recherche, nous nous sommes limités, dans cette dernière, à l'utilisation d'un système de pondération simple. Cependant, il existe, dans la littérature, plusieurs autres algorithmes de pondération susceptible d'améliorer la qualité des recommandations de notre système (Amati et Joost Van Rijsbergen, 2002 ; Manning, Raghavan et Schtze, 2008). Nous envisageons d'examiner ces différents algorithmes de pondération dans nos futurs travaux.

5.2.3. Discrimination du voisinage

L'originalité de notre approche réside dans l'utilisation d'une technique de sélection préalable des voisins utilisés dans le calcul de la recommandation. Cette technique de discrimination, illustrée par la figure 5.1, consiste à sélectionner, lors de la formation du voisinage pour le calcul des prédictions, un sous-ensemble de voisin constitué par un groupe de voisins à forte similarité et un groupe de voisins à faible similarité. Notre hypothèse est que, d'une part, les voisins à moyenne similarité constituent du bruit pour le système de recommandation. D'autre part, nous croyons que l'information pertinente pour le système de recommandation ne se limite pas uniquement aux voisins à forte similarité (Deshpande et Karypis, 2004), mais aussi aux voisins à faible similarité. De même, nous pensons que cette approche nous permettra, d'une part, de réduire la complexité computationnelle du système de recommandation et d'autre part, d'améliorer la qualité de la recommandation.

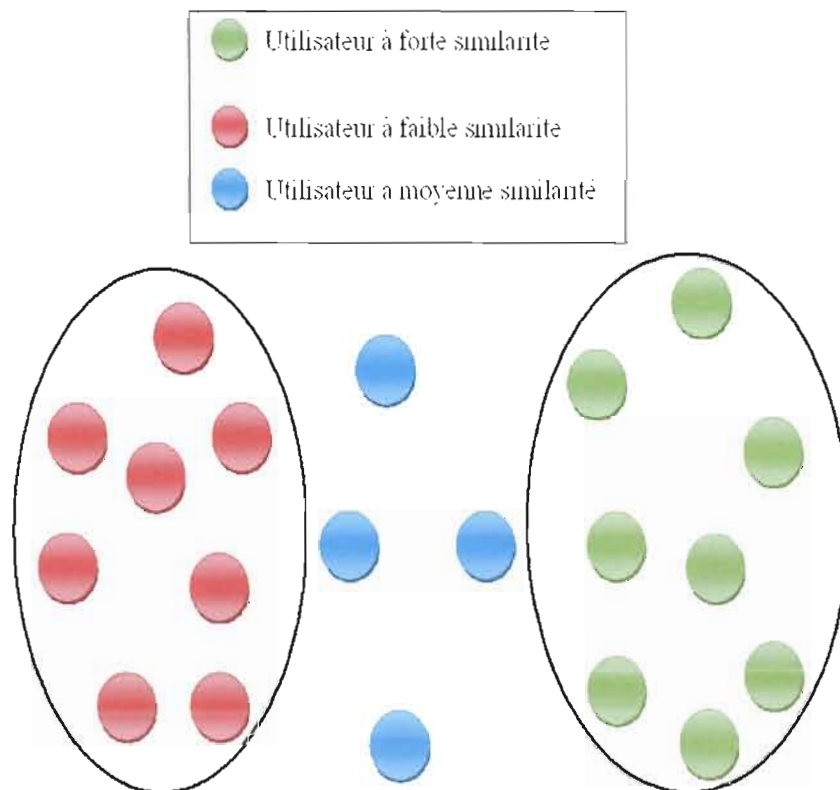


Figure 5.1 Technique de discrimination du voisinage

5.2.4. Calcul de la similarité entre utilisateurs

La mesure de similarité entre les profils est calculée à l'aide du *coefficient de corrélation de Pearson*, du *vecteur cosinus* ou des *moindres carrés*.

Le coefficient de corrélation de Pearson:

$$w(u_i, u_k) = \frac{\sum_{r \in R_i \cap R_k} (eval(u_i, r) - \bar{v}) (eval(u_k, r) - \bar{v})}{\sqrt{\sum_{r \in R_i \cap R_k} (eval(u_i, r) - \bar{v})^2} \sqrt{\sum_{r \in R_i \cap R_k} (eval(u_k, r) - \bar{v})^2}}$$

Les moindres carrés:

$$w(u_i, u_k) = \frac{1}{\sum_{r \in R_i \cap R_k} (eval(u_i, r) - eval(u_k, r))^2}$$

Le vecteur cosinus:

$$w(u_i, u_k) = \frac{\sum_{r \in R_i \cap R_k} eval(u_i, r) eval(u_k, r)}{\sqrt{\sum_{r \in R_i} eval(u_i, r)^2} \sqrt{\sum_{r \in R_k} eval(u_k, r)^2}}$$

Une étude comparative des différentes mesures de similarité est fournie par Zaier et al. (2007). Notre algorithme utilise la métrique de Pearson.

5.2.5. Calcul des prédictions

Dans notre approche, l'évaluation de l'utilisateur u_i , prédite par notre algorithme de filtrage, sur l'article r_j est calculée en utilisant l'équation suivante (Zaier, Godin et Faucher, 2007, 2008b) :

$$\overline{eval}(u_i) + k * \sum_{u \in U} (w(u_i, u) * (eval(u, r_i) - \overline{eval}(u)))$$

Où $\overline{eval}(u_i)$ et $\overline{eval}(u)$ sont les évaluations moyennes respectives de l'utilisateur u_i et l'utilisateur u . $eval(u, r_i)$ est l'évaluation faite par l'utilisateur u sur l'article r_i , et le poids $w(u_i, u)$ est la similarité, entre l'utilisateur u_i et l'utilisateur u . Dans notre cas, nous avons pris $k = \frac{1}{\sum_{u \in U} w(u_i, u)}$.

5.3. Architecture du système

Une grande partie des recherches sur le filtrage collaboratif ont été consacrées aux systèmes de recommandation centralisés. Les utilisateurs de ces systèmes ne peuvent interagir qu'à travers ce serveur central. En effet, le serveur rassemble l'information sur les préférences des utilisateurs par leur interaction, implicite ou explicite, avec le serveur. En se basant sur ces profils, le serveur fournit ses recommandations aux utilisateurs. La figure 5.2 décrit l'architecture générale de systèmes centralisés de recommandation.

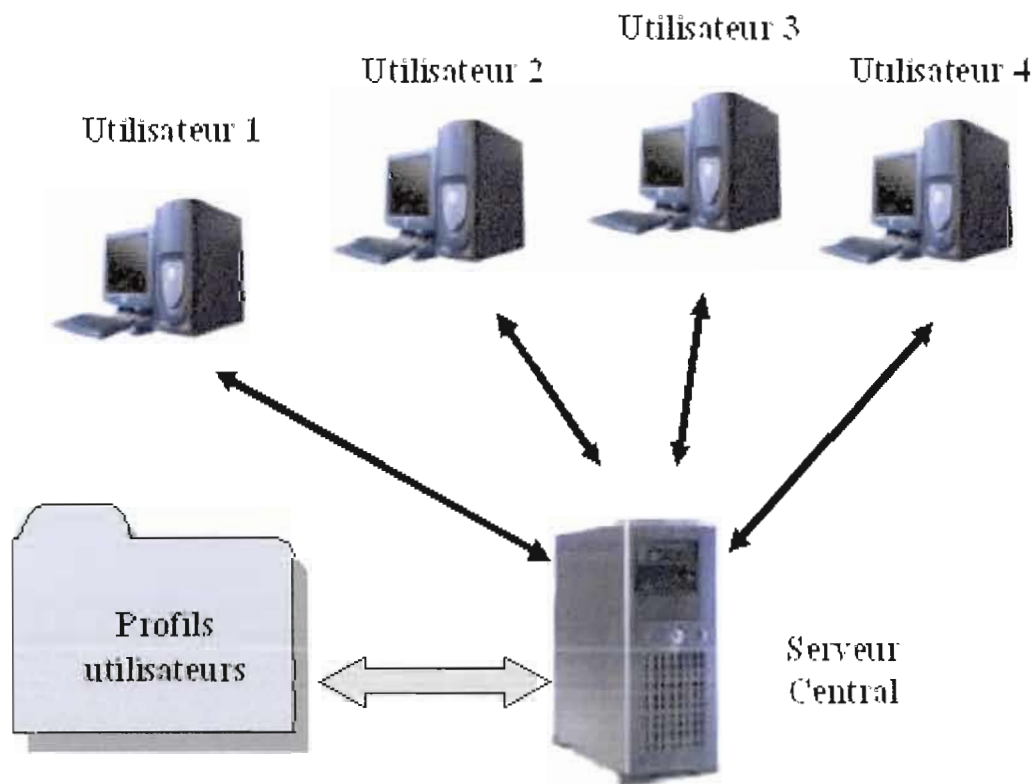


Figure 5.2 Architecture générale d'un système de recommandation centralisé

Le chapitre 3 a mis en évidence plusieurs inconvénients des systèmes de recommandation centralisés. En effet, un système de recommandation centralisé doit stocker une quantité considérable de données pour nombre important d'utilisateurs. De ce fait, le système doit mobiliser de plus en plus de ressources coûteuses (puissance de calcul, mémoire, bande passante, etc.). En outre, si le serveur central du système de recommandation tombe en panne ou il n'est plus accessible, l'utilisateur ne peut plus utiliser le service : déni de service (« *Denial of Service* » : *DoS*). Par ailleurs, la confidentialité des informations relatives aux utilisateurs constitue un autre problème. En effet, les politiques de sécurité et de protection des renseignements privés sont définies par le serveur du système de recommandation. Nous avons donc opté pour une architecture décentralisée pour notre système de recommandation.

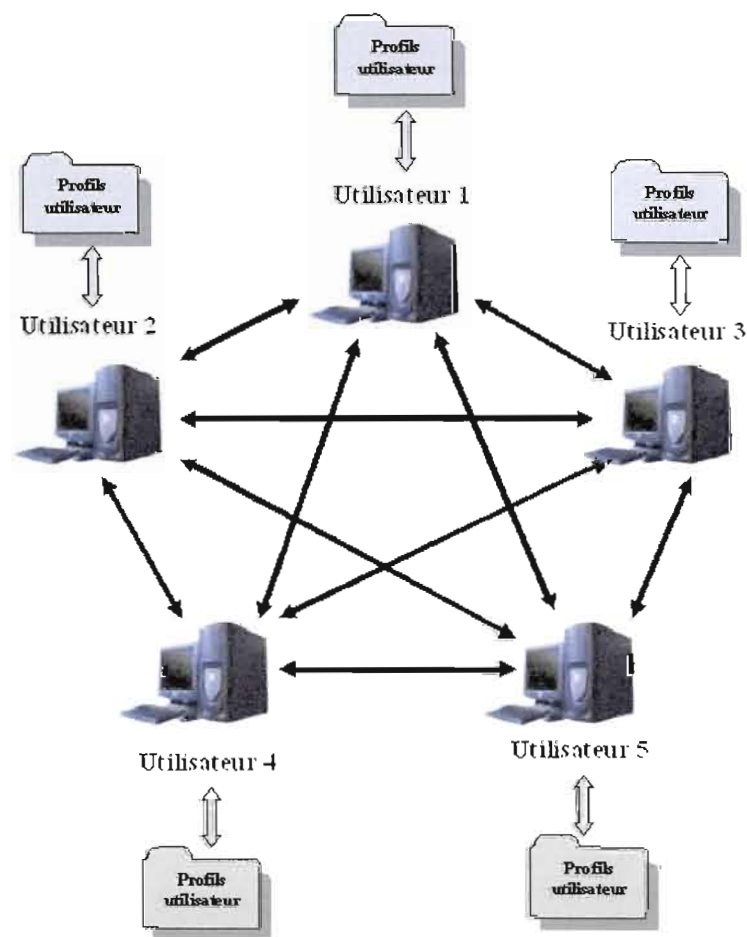


Figure 5.3 Architecture générale de systèmes décentralisés de recommandation

Ce système de recommandation distribué se chargera de faire communiquer plusieurs utilisateurs entre eux par biais d'une architecture pair-à-pair pour former des groupes d'utilisateurs selon la métrique de similarité. Cependant, il y a deux propriétés qui affectent l'architecture d'un tel système : où est stockée l'information et où sont calculées les recommandations. Pour ce qui nous concerne, nous avons opté pour système de recommandation purement décentralisé, illustré par la figure 5.3. Ainsi, l'information est stockée et les recommandations sont calculées au niveau des utilisateurs. Plus précisément, chaque utilisateur du système correspond à un pair. Chaque pair dispose donc d'un identifiant unique et de ses profils. Ainsi, les données sont stockées sur la machine locale afin de préserver la vie privée. Pour ce faire, nous avons opté pour le protocole pair-à-pair *Chord*.

Le protocole de nouvelle génération *Chord*, introduit dans le chapitre 3, est totalement distribué et fortement évolutif (Miller, Konstan et Riedl, 2004). Une étude comparative a montré que dans le cadre d'un système de recommandation le réseau *Chord* est la solution la plus optimale (Miller, Konstan et Riedl, 2004). Les utilisateurs du réseau en anneau *Chord* disposent de fonctions de recherche et de génération de clés de hachage pour publier ou récupérer des données. Dans notre système de recommandation, chaque pair gère plusieurs types de données :

- ✓ une liste, désignée *IDG*, des identifiants des pairs appartenant au groupe,
- ✓ une liste, nommée *IDS*, des identifiants des pairs qui ont été sélectionnés par notre algorithme de discrimination (dont la similarité appartient aux extrêmes),
- ✓ une liste, appelée *IDA*, des identifiants des pairs ayant ajouté le profil de l'utilisateur courant à leur profil de groupe,
- ✓ les profils sélectionnés,
- ✓ un profil public qui pourra être échangé à travers le réseau par le biais du protocole *Chord*.

Le pair courant demande le profil public de tous les pairs qu'il peut contacter sur le réseau. L'identifiant des pairs en question est ajouté à la liste *IDG*. Puis, pour chacun de ces profils, nous calculons le degré de similarité avec le profil du pair courant. Si le résultat est accepté par l'algorithme de discrimination, l'identifiant du pair en question est ajouté à la liste *IDS* et le profil correspondant est inclus dans les profils sélectionnés de l'utilisateur courant. La liste *IDS* et les profils sélectionnés vont servir dans le calcul des recommandations. Si le profil public de l'utilisateur courant est modifié, l'utilisateur courant doit contacter tous les pairs dont l'identifiant est dans sa liste *IDA*. Chacun de ces pairs recalcule la métrique de similarité. Si le résultat est accepté par l'algorithme de discrimination, l'identifiant de ce dernier est ajouté à la liste *IDS* et son profil est intégré dans les profils sélectionnés. Sinon, l'identifiant et le profil sont respectivement supprimés de la liste *IDS* et des profils sélectionnés.

5.4. Plateforme multi-agent

Comme nous l'avons vu dans le chapitre 4, il existe un lien naturel entre les systèmes pair-à-pair et les systèmes multi-agents. Ainsi, nous considérons, dans le cadre de notre approche, notre système distribué de filtrage collaboratif comme un ensemble d'agents indépendants, un agent pour chaque pair du système. Le système est constitué ainsi d'une population d'agents autonomes en interaction, chaque agent possédant des capacités de communication, de coordination et de collaboration. Chaque agent possède une vue, soit directe, soit indirecte, sur les autres agents. Cette vue reflète l'importance sociale de l'agent et évolue en fonction de ses interactions.

Dans le cadre de notre approche, nous avons opté pour un système multi-agents composé par des agents réactifs, introduits dans le chapitre 4 où, chaque agent a un comportement réactif simple basé sur la coopération. L'hypothèse derrière ce choix est qu'un tel système réunit les conditions nécessaires pour permettre l'émergence d'une organisation au sein du système et donc, également, l'émergence de sa fonction globale. Ce comportement global émerge du comportement des agents qui composent le système et de leurs interactions. Ainsi, des comportements sociaux complexes peuvent émerger au niveau du regroupement et de la spécialisation des pairs, au niveau du partage des ressources, etc. Par exemple, nous pouvons

avoir l'émergence de comportements de coopération ; l'émergence de pairs experts (socialement plus importants) : une hiérarchie ; l'émergence de comportement de tromperie ; etc.

Chapitre VI

Évaluation de l'approche proposée

« Examine si ce que tu promets est juste et possible, car la promesse est une dette. »

- Confucius (-551 - -479)

6.1. Introduction

Au chapitre précédent, nous avons exposé notre approche de système de recommandation. Nous y avons intégré un modèle hybride de filtrage d'informations et une architecture décentralisée. Ce faisant, on souhaite que notre système soit en mesure de mieux répondre à sa tâche. Plus précisément, l'effet escompté de l'utilisation de cette nouvelle approche est une amélioration des recommandations. Pour valider cette hypothèse, il faut répondre aux quatre questions suivantes :

- ✓ Notre modèle amène-t-il une amélioration de la performance et de la qualité des documents résultants de la méthode de filtrage de l'information ?
- ✓ Est-ce que l'utilisateur tire profit du système ?
- ✓ Est-ce qu'il y a un recouvrement plus complet de l'espace d'information ?
- ✓ Est-ce que notre système offre un bon rapport qualité/performance ? En d'autres termes, est-ce que nous avons pu trouver un bon compromis entre la qualité des documents et la performance du système (temps de réponse, consommation de ressources système, etc.) ?

Pour répondre à ces questions, ce chapitre a pour but de spécifier d'abord et avant tout la méthodologie d'évaluation mise en place, puis de présenter les résultats des différentes expérimentations menées et de discuter des interprétations qu'on peut en faire.

6.2. Méthodologie d'évaluation

L'évaluation de notre modèle est capitale pour la quantification de la profitabilité des recommandations fournies par le système, en comparaison aux évaluations réelles. Cependant, l'évaluation « en ligne » (réelle), avec l'aide d'utilisateurs réels, n'est pas à l'heure actuelle, une option réaliste dans le cadre de nos expérimentations. Étant donné que, d'une part, nous ne possédons pas les ressources (groupe réel d'utilisateurs, matériels, temps, etc.) pour le faire. D'autre part, dans un environnement de simulation, il est plus facile de surveiller tous les paramètres des expérimentations.

De ce fait, nous allons procéder à des expérimentations « hors ligne » (simulations) pour évaluer notre système de recommandation. Dans ce but, il faut définir, selon les règles de l'art du domaine des corpus sur lesquels notre système de recommandation peut être appliqué ainsi que des métriques pour évaluer notre système de recommandation, les différentes méthodes utilisées dans nos expériences et finalement mettre au point une procédure d'évaluation.

6.2.1. Corpus d'évaluation

6.2.1.1. Corpus pour l'évaluation des systèmes de recommandation

Les chercheurs qui travaillent dans le domaine du filtrage collaboratif de l'information utilisent différents corpus de plusieurs domaines d'application pour évaluer la performance des algorithmes de recommandation (Herlocker *et al.*, 2004). On trouvera une revue de la littérature sur les corpus d'évaluation pour les systèmes de recommandation dans Zaier *et al.* (2008a).

Corpus MovieLens. Ces derniers sont les plus populaires dans l'évaluation des systèmes de recommandation. En effet, les corpus *MovieLens* (www.grouplens.org) ont déjà fait

l'objet de plusieurs travaux de recherche (Miller, Konstan et Riedl, 2004 ; Zaier, Godin et Faucher, 2007, 2008b). Publiquement disponibles, ils contiennent des évaluations explicites au sujet de films, des informations démographiques sur les utilisateurs (âge, genre, métier, code postal) et une courte description des films (titre, année de production, genres). Ainsi, ces corpus, à très forte densité, sont constitués d'évaluations de films, sur une échelle de 1 à 5, faites par 943 utilisateurs anonymes sur 1682 films. De plus, cette collection est subdivisée en plusieurs corpus d'évaluation.

Corpus Jester. Ces corpus, proposés par *Ken Goldberg* du site Web de recommandation de blagues *Jester* (www.ieor.berkeley.edu/~goldberg/jester-data/), contiennent des évaluations, de 100 blagues, faites par 73.496 utilisateurs anonymes. Les notes sont des valeurs réelles qui varient entre -10.00 et +10.00 (Canny, 2002a). Ces corpus, comme pour les corpus *MovieLens*, ont une très forte densité. En effet, un grand nombre d'utilisateurs ont évalué, à peu près, toutes les blagues.

Cette collection est divisée en trois corpus différents :

- ✓ Jester-data-1 : Contient les évaluations de 24.983 utilisateurs qui ont évalué 36 blagues ou plus.
- ✓ Jester-data-2 : Contient les notes de 23.500 utilisateurs qui ont évalué 36 blagues ou plus.
- ✓ Jester-data-3 : Contient les données de 24.938 utilisateurs qui ont évalué entre 15 et 35 blagues.

Corpus Netflix. Le corpus *Netflix* (www.netflix.com) est actuellement l'objet d'un nombre considérable d'études. En 2006, le spécialiste de la location de films en ligne lance le *Netflix Prize* (www.netflixprize.com), avec un million de dollars pour qui augmentera de dix pour cent la qualité de leur système de recommandation *Cinematch*TM. Ce dernier prédit si un spectateur va aimer ou non un film en fonction des films qu'il a appréciés (ou pas) par le passé. À cet effet, *Netflix* a proposé un gros corpus comprenant plus de 100 millions d'évaluations de 17.770 films, sur une échelle de 1 à 5, fournies par 480.189 utilisateurs

anonymes. De plus, ce corpus comporte une courte description des films (titre et année de production) (Bell et Koren, 2007b, 2007a, 2007c ; Bell, Koren et Volinsky, 2007c, 2007a, 2007b).

Corpus BookCrossing. Le corpus BookCrossing a été recueilli en 2004 par *Cao-Nicolas Ziegler* (www.informatik.uni-freiburg.de/~ciegler/BX/), à partir de la communauté BookCrossing (www.bookcrossing.com). Il est composé de 278.858 utilisateurs anonymes qui ont fourni 1.149.780 évaluations sur 271.379 livres, notées sur une échelle de 1 à 10. Ce corpus, comme pour le corpus de MovieLens, contient des informations démographiques sur les utilisateurs (âge et ville) et une courte description des livres (titre du livre, auteur du livre, année de publication, éditeur) (Ziegler, 2005 ; Ziegler *et al.*, 2005).

6.2.1.2. Présentation des corpus utilisés

Une attention particulière est portée quant au choix des corpus. Une de nos préoccupations est de choisir des corpus facilement accessibles et disponibles, ayant déjà été utilisés dans le passé et étant susceptibles de l'être encore dans le futur. Un autre aspect assez important est la taille de ces corpus. Pour que nos résultats soient significatifs, nos bases de travail devaient contenir un nombre suffisant d'utilisateurs, d'article à évaluer, et d'évaluations. Une récente étude, sur la distribution des corpus d'évaluation des systèmes de recommandation, montre que cette dernière pouvait influencer la performance des algorithmes (Zaier, Godin et Faucher, 2008a). Il est donc préférable de choisir un corpus qui suit une distribution réelle et ainsi, obtenir des performances qui reflètent la réalité (Anderson, 2004, 2006 ; Elberse, 2008).

Nous avons par conséquent opté pour les corpus *MovieLens*, qui présentent toutes ces propriétés. En effet, ces corpus sont à la fois facilement exploitables et publiquement disponibles (Herlocker *et al.*, 2004 ; Miller, Konstan et Riedl, 2004 ; Zaier, Godin et Faucher, 2007, 2008b). De plus, ces corpus sont assez grands et denses. Dans ces corpus, il y a 10.000 évaluations faites par 943 utilisateurs sur 1682 films (seulement les utilisateurs qui ont évalué au moins 20 films sont considérés). En outre, cette collection contient des informations décrivant à la fois les utilisateurs et les films. De même, elle a été l'objet de très

nombreuses investigations empiriques (Zaier, Godin et Faucher, 2007, 2008b). Enfin, ces corpus suivent une distribution réelle (Zaier, Godin et Faucher, 2008a).

Plusieurs corpus de la collection *MovieLens* ont été choisis pour nos expériences. Les cinq premiers sont composés de 10.000 évaluations comme ensemble d'apprentissage (par exemple *U1.BASE*) et 20.000 prédictions comme ensemble de test (par exemple *U1.TEST*). Ces collections s'appelleront respectivement *U1* à *U5*. Les deux corpus restant sont constitués de 10.000 évaluations comme ensemble d'apprentissage (par exemple *UA.BASE*) et 9.430 prévisions (exactement 10 estimations par utilisateur.) comme ensemble de test (par exemple *UA.TEST*). Ces collections s'appelleront respectivement *UA* et *UB*. Chacun de ces corpus a été transformé en une matrice utilisateur-film, notée *UserVSItem* (voir section 6.2.3), avec 943 lignes (Utilisateurs) et 1682 colonnes (films).

6.2.2. Métriques d'évaluation

6.2.2.1. Catégories principales de métriques d'évaluation pour les systèmes de recommandation

La revue de la littérature, la plus exhaustive et la plus complète, sur les métriques d'évaluation pour les systèmes de recommandation est fournie par Herlocker et al. (2004). Cette étude montre que les métriques d'évaluation se divisent généralement en 3 catégories principales. En premier lieu, il y a les fonctions d'erreurs qui évaluent la prédiction faite par le système en la comparant à l'évaluation réelle. Ensuite, il existe les métriques d'aide à la décision qui jugent de l'efficacité du choix des documents pertinents. Enfin, d'autres métriques ont été proposées pour évaluer des critères qui ne sont pas liés aux prédictions (Herlocker *et al.*, 2004 ; McNee, Riedl et Konstan, 2006 ; Zenebe et Norcio, 2005 ; Ziegler, 2005).

Fonctions d'erreurs. Ces dernières sont les mesures les plus utilisées dans la littérature. Elles évaluent la qualité des prédictions générées par le système de recommandation (Herlocker *et al.*, 2004). Il existe plusieurs types de fonctions d'erreur qui consistent à mesurer la « distance » moyenne entre les prévisions et les observations correspondantes. Ainsi, une valeur proche de 0 indique des prédictions parfaites et une valeur avoisinante de 1

désigne de mauvaises prédictions (Adomavicius et Tuzhilin, 2005 ; Herlocker, 2000 ; Herlocker *et al.*, 2004 ; Ziegler, 2005 ; Ziegler *et al.*, 2005). Pour mieux illustrer les mesures qui vont suivre, on suppose que w_i représente la valeur observée et r_i représente la valeur prédite pour un ensemble de B_i prévisions. Ainsi, on peut calculer :

- ✓ L'erreur absolue moyenne (*MAE* : « *Mean Absolute Error* ») : cette mesure est la fonction d'erreur la plus populaire. Elle évalue la qualité des prédictions fournies par le système de recommandation. À cet effet, le *MAE* mesure la déviation absolue moyenne entre une estimation prévue et l'estimation vraie de l'utilisateur (Canny, 2002b ; Lemire et MacLachlan, 2005 ; Miller, Konstan et Riedl, 2004 ; Sarwar *et al.*, 2000a ; Zaier, Godin et Faucher, 2007, 2008b).

$$|\bar{E}| = \frac{\sum_{b_k \in B_i} |r_i(b_k) - w_i(b_k)|}{|B_i|}$$

- ✓ L'erreur moyenne quadratique (*MSE* : « *Mean Squared Error* ») : cette fonction d'erreur est appréciée pour sa grande sensibilité à l'erreur en comparaison au *MAE*. En effet, on peut voir que la distance, entre la prévision et l'observation, est élevée au carré (Ziegler *et al.*, 2005).

$$|\bar{E}| = \frac{\sum_{b_k \in B_i} (r_i(b_k) - w_i(b_k))^2}{|B_i|}$$

- ✓ La racine de l'erreur moyenne quadratique (*RMSE* : « *Root Mean Squared Error* ») : dans la littérature, le *RMSE* est largement utilisé, à la place du *MSE*, pour évaluer les systèmes de recommandation. Il est utilisé par le fameux concours *Netflix Prize* (<http://www.netflixprize.com/>) pour identifier les meilleurs algorithmes de filtrage (Bell et Koren, 2007b, 2007a, 2007c ; Bell, Koren et Volinsky, 2007c, 2007a, 2007b).

$$|\bar{E}| = \sqrt{\frac{\sum_{b_k \in B_i} (r_i(b_k) - w_i(b_k))^2}{|B_i|}}$$

✓ L'erreur absolue moyenne normalisée (*NMAE* : « *Normalized Mean Absolute Error* ») : cette mesure est définie comme le MAE divisé par la gamme des valeurs possibles des évaluations observées. Certains chercheurs préfèrent utiliser le *NMAE* plutôt que le MAE (Canny, 2002a ; Lemire, 2005). Son utilisation se justifie quand on doit procéder à des expérimentations utilisant différents corpus d'évaluation. En effet, cette mesure fournit une valeur normalisée du *MAE* facilitant la comparaison pour des échelles de notation différentes (Goldberg *et al.*, 2001).

$$|\bar{E}| = \frac{\frac{\sum_{b_k \in B_i} |r_i(b_k) - w_i(b_k)|}{|B_i|}}{\tilde{r}_{max} - \tilde{r}_{min}}$$

✓ L'erreur absolue utilisateur moyenne (*MAUE* : « *Mean Absolute User Error* ») : cette mesure, introduite par Massa et Avesani (2004), calcule d'abord l'erreur absolue moyenne pour chaque utilisateur ; ensuite, elle calcule la moyenne de ces erreurs utilisateurs sur l'ensemble de tous les utilisateurs. Ainsi, chaque utilisateur aura le même poids dans le calcul de l'erreur absolue utilisateur moyenne (Massa et Avesani, 2004, 2007a).

Métriques d'aide à la décision. Ces métriques sont, généralement, utilisées dans l'évaluation des systèmes de repérage d'informations. À l'opposé des fonctions d'erreurs, les métriques d'aide à la décision n'évaluent pas la qualité de la prédiction, mais la pertinence des documents. Ainsi, ces mesures prennent seulement en considération la pertinence du document (Adomavicius et Tuzhilin, 2005 ; Haase *et al.*, 2005 ; Herlocker, 2000 ; Ziegler, 2005 ; Ziegler *et al.*, 2005). Pour mieux illustrer les mesures qui vont suivre, on prend pour point de départ le tableau suivant de classes de documents.

Tableau 6.1 Les classes de documents

	Choisi	Refusé	Total
Pertinent	N_{pc}	N_{pr}	N_p
Non pertinent	N_{nc}	N_{nr}	N_n
Total	N_c	N_r	N

On définit à partir des classes de ce tableau les mesures suivantes :

- ✓ Précision (« *Precision* ») : cette mesure représente la probabilité qu'un article choisi soit pertinent (Sarwar *et al.*, 2000b).

$$P = \frac{N_{pc}}{N_c}$$

- ✓ Rappel (« *Recall* ») : cette métrique calcule la probabilité qu'un article pertinent soit choisi (Miller, Konstan et Riedl, 2004 ; Sarwar *et al.*, 2000b ; Zaier, Godin et Faucher, 2007, 2008b).

$$R = \frac{N_{pc}}{N_p}$$

- ✓ F_1 : cette mesure est une façon courante de combiner le rappel et la précision dans une seule métrique afin de faciliter la comparaison (Huang, Chen et Zeng, 2004 ; Montaner, 2003 ; Sarwar *et al.*, 2000b).

$$F_1 = \frac{2PR}{P + R}$$

- ✓ *Breese score* : Breese *et al.* (1998) proposent une extension à la métrique de *Rappel* appelée *Breese score* ou *Rappel pondéré*. L'idée principale sur laquelle repose cette métrique dérive du fait que l'utilité d'une recommandation se base essentiellement sur la probabilité que cette dernière soit consultée. Ainsi, cette mesure tient compte de l'ordre de présentation des recommandations en leur assignant différents poids. Pour ce faire, la métrique pénalise les n premières recommandations incorrectes, déterminées à partir d'une fonction appelée Breese-points, en leur assignant un poids fort et en diminuant de plus en plus ce dernier au fur et à mesure qu'on descend dans la liste de recommandations (Stritt, Tso et Schmidt-Thieme, 2006 ; Ziegler *et al.*, 2005).
- ✓ La courbe *ROC* (« *Receiver Operating Characteristic* ») : une courbe ROC permet de comparer des algorithmes de recommandation indépendamment de la de la qualité des prédictions. ROC mesure le point auquel un système de filtrage de l'information peut distinguer avec succès les articles pertinents des articles non pertinents. De plus, cette mesure évalue l'ordre de présentation des recommandations (rang). Dans ce but, cette courbe met en relation dans un graphique, illustré par la figure 6.1, les taux de faux positifs (en abscisse) et les taux de vrais positifs (en ordonnée) (Good *et al.*, 1999 ; Melville, Mooney et Nagarajan, 2002 ; Rojsattarat et Soonthornphisaj, 2003 ; Schein *et al.*, 2002 ; Van

Setten *et al.*, 2004 ; Ziegler *et al.*, 2005). Une amélioration de la courbe *ROC*, notée *CROC*, a été proposée par Shein *et al.* (2005).

- ✓ *NDPM* (« *Normalized Distance-based Performance Measure* ») : rarement utilisée, cette mesure, comme pour la courbe *ROC*, évalue l'ordre de présentation des recommandations. Plus précisément, cette métrique mesure la « distance » entre le rang de l'article dicté par l'évaluation de l'utilisateur et le rang prédit par le l'algorithme de recommandation (Balabanovi et Shoham, 1997).

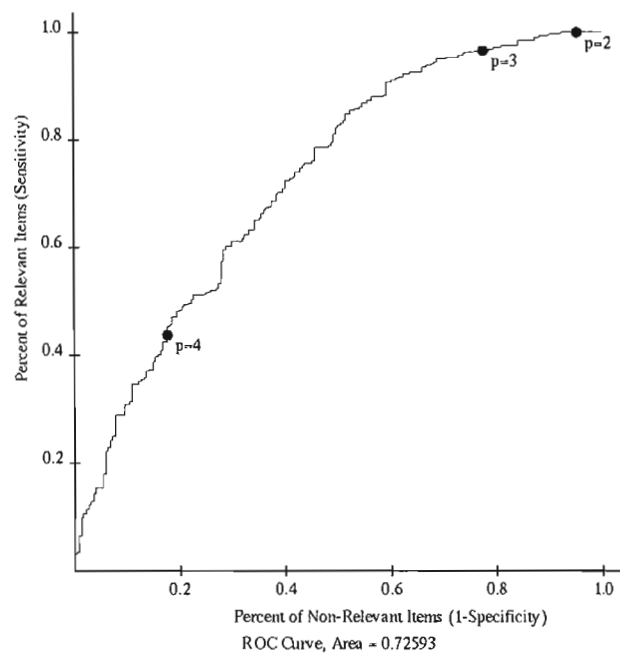


Figure 6.1 La courbe *ROC* (Herlocker *et al.*, 2004)

Autres métriques. Bien que les fonctions d'erreurs et les métriques d'aide à la décision soient importantes dans l'évaluation de la performance des systèmes de recommandation, il existe des aspects qui ne sont pas pris en compte par ces dernières (McNee, Riedl et Konstan, 2006). Ainsi, de nouvelles mesures d'évaluation ont été proposées (Adomavicius et Tuzhilin, 2005 ; Herlocker *et al.*, 2004 ; Miller, 2003 ; Ziegler, 2005):

- ✓ **Couverture :** la couverture mesure le pourcentage des articles pour lesquels le système a pu calculer des recommandations. Selon Miller *et al.* (2004), il existe deux façons de calculer la métrique de couverture. La première calcule le

pourcentage global des articles pour lesquels le système de recommandation a pu calculer des prédictions. La deuxième évalue le pourcentage des articles choisis par l'utilisateur pour lesquels le système de recommandation a pu calculer des prédictions (Adomavicius et Tuzhilin, 2005 ; Good *et al.*, 1999 ; Herlocker, 2000 ; Massa et Avesani, 2004 ; Middleton, Shadbolt et De Roure, 2004 ; Zaier, Godin et Faucher, 2007, 2008b).

- ✓ Taux d'apprentissage : les algorithmes d'apprentissage sont une composante importante des systèmes de recommandation. Leur performance varie en fonction de la quantité de données d'apprentissage disponibles. Si la quantité de données d'apprentissage augmente, la qualité des recommandations devrait augmenter. Donc, cette mesure évalue la qualité des recommandations par rapport à la quantité de données d'apprentissage disponibles (Herlocker *et al.*, 2004).
- ✓ Nouveauté et hasard (« *Novelty and Serendipity* ») : pour illustrer cette mesure, Ziegler (2005) prend pour exemple l'achat de bananes dans une épicerie. Dû à la popularité élevée de ces dernières, la plupart des personnes les achètent intuitivement. Donc, ces personnes n'ont pas besoin d'une recommandation supplémentaire puisqu'elles savent déjà qu'elles vont acheter des bananes. Cette métrique mesure ainsi les recommandations non évidentes, tout en pénalisant celles évidentes (Herlocker *et al.*, 2004 ; Terveen et Hill, 2001).
- ✓ Satisfaction des utilisateurs : les travaux de Cosley *et al.* (2003) montrent que la satisfaction de l'utilisateur diminue quand un nombre significatif d'erreurs est produit par le système de recommandation. Ainsi, il peut être possible d'établir, très facilement, le niveau de satisfaction des utilisateurs (Cosley *et al.*, 2003 ; Miller, Konstan et Riedl, 2004).
- ✓ Similarité moyenne : cette mesure, introduite par Miller (2003), s'intéresse à la qualité du voisinage. Plus précisément, la similarité moyenne est une mesure qui évalue la similarité moyenne du groupe d'utilisateurs choisis pour le calcul de la recommandation finale (Miller, Konstan et Riedl, 2004).

- ✓ Ressources utilisées : cette dernière s'intéresse à la complexité de l'algorithme de recommandation. En effet, cette mesure calcule les ressources consommées par l'algorithme. Plus précisément, les ressources considérées peuvent être le temps de réponse, la consommation de ressources systèmes, etc. (Miller, Konstan et Riedl, 2004 ; Miller, 2003).

6.2.2.2. Présentation des métriques utilisées

Comme suggéré par Herlocker et *al.* (2004) et validé Miller et *al.* (2004), nous avons opté pour les métriques suivantes :

Erreur absolue moyenne. Nous avons choisi cette mesure pour évaluer le niveau d'exactitude des recommandations fournies par le système.

$$|\bar{E}| = \frac{\sum_{b_k \in B_i} |r_i(b_k) - w_i(b_k)|}{|B_i|}$$

Où w_i représente la valeur observée et r_i représente la valeur prédite pour un ensemble de B_i prévisions.

Rappel. Nous avons choisi cette mesure pour évaluer la qualité des recommandations faites par le système.

$$R = \frac{N_{pc}}{N_p}$$

Où N_{pc} est le nombre d'articles pertinents prédis par le système et N_p est le nombre total d'articles pertinents.

Couverture. Cette mesure évalue la couverture des recommandations.

$$C = \frac{N_p - N_{pn}}{N_p}$$

Où N_{pr} est le nombre d'articles pertinents que le système ne peut pas prédire et N_p est le nombre total d'articles pertinents.

6.2.3. Méthodes utilisées dans nos expériences

Cette section décrit en détail les trois approches de filtrage utilisées dans nos expériences pour l'évaluation de l'impact de notre algorithme de discrimination du voisinage. Nous n'avons pas inclus l'approche de filtrage basée sur contenu, présentée dans la section 2.3.1, car elle n'est pas affectée par notre algorithme (discrimination du voisinage). Nous avons défini, pour ces approches, différentes matrices utilisées dans le calcul des prédictions (*UserVSItem*, *ItemVSTerm*, *UserVSTerm*, et *UserVSDem*). Par exemple, la matrice *UserVSItem* représente l'évaluation faite par un utilisateur donné sur un article donné.

Filtrage collaboratif. L'approche de filtrage collaboratif, introduite dans la figure 6.2, recommande des articles à un utilisateur donné en se basant sur les évaluations d'autres utilisateurs. Une fois que le voisinage d'utilisateurs est formé, l'ensemble des évaluations est représenté sous forme d'une matrice *UserVSItem* avec des utilisateurs pour lignes et des articles pour colonnes. Dans cette approche, le degré de similarité, entre un utilisateur u_i et un utilisateur u_j , est calculé en utilisant le coefficient de corrélation de Pearson en se basant sur les évaluations faites par les utilisateurs. Le résultat de cette étape est une matrice *UserVSUser* qui définit la corrélation entre tous les utilisateurs. Dans notre travail, l'évaluation de l'utilisateur u_i , prédite par l'algorithme de filtrage, sur l'article r_j est calculée en utilisant l'équation suivante (Zaier, Godin et Faucher, 2007, 2008b) :

$$\overline{eval}(u_i) + k * \sum_{u \in U} (w(u_i, u) * (eval(u, r_i) - \overline{eval}(u)))$$

Où $\overline{eval}(u_i)$ et $\overline{eval}(u)$ sont les évaluations moyennes respectives de l'utilisateur u_i et l'utilisateur u . $eval(u, r_i)$ est l'évaluation faite par l'utilisateur u sur l'article r_i , et le poids $w(u_i, u)$ est la similarité, entre l'utilisateur u_i et l'utilisateur u , données par la matrice *UserVSUser*. Dans notre cas, nous avons pris $k = \frac{1}{\sum_{u \in U} w(u_i, u)}$.

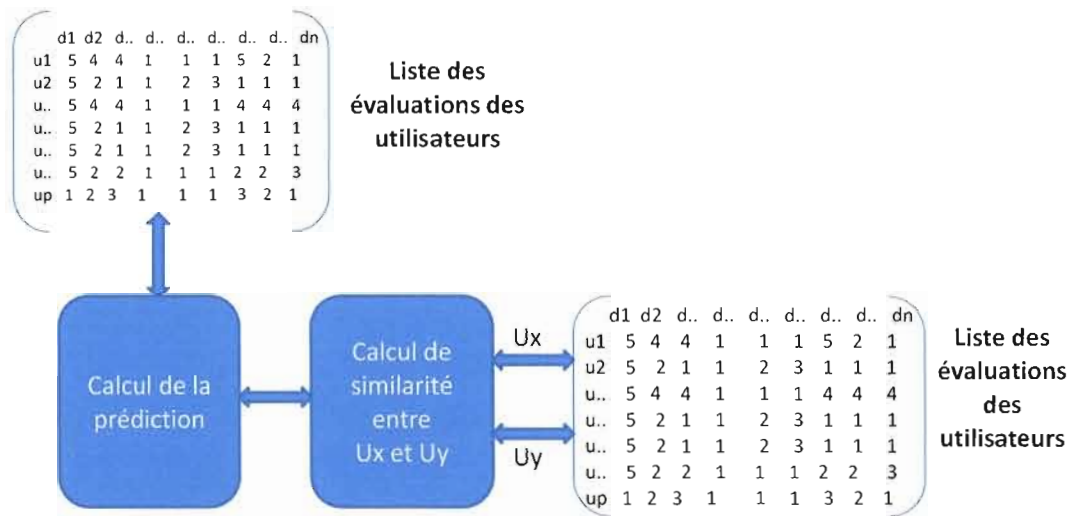


Figure 6.2 Filtrage collaboratif

Filtrage hybride. Afin d'évaluer l'approche de filtrage hybride, exposée dans la figure 6.3, nous avons téléchargé les descriptions de films (pilote, acteurs, pays) du site web *IMBD* (*Internet Movie DataBase*, www.imdb.com) et nous les avons associées aux données existantes dans les corpus *MovieLens* introduits dans la section 6.2.1 (titre, année de production, genres). Cette étape est illustrée par la figure 6.4. En outre, nous avons calculé le *TF-IDF* des termes par document représenté sous forme d'une matrice *ItemVSTerm*. Alors, nous calculons la matrice *UserVSTerm*, qui identifie le profil d'utilisateur, par le biais de l'algorithme de *Rocchio* (Adomavicius et Tuzhilin, 2005 ; Zaier, Godin et Faucher, 2007, 2008b):

Où, pour un utilisateur donné j , C est l'ensemble des articles pertinents identifiés à partir de la matrice *UserVSTerm*, n est le nombre total d'articles, n_c est le nombre d'articles pertinents, $w_{i,j}$ est le poids existant, et le $x_{i,j}$ est le *TF-IDF* de chaque terme de l'article i de la matrice *ItemVSTerm*. Dans notre expérience, nous avons pris les coefficients $\alpha=0$, $\beta=1$ et $\gamma=1$. Cette étape est illustrée par la figure 6.5.

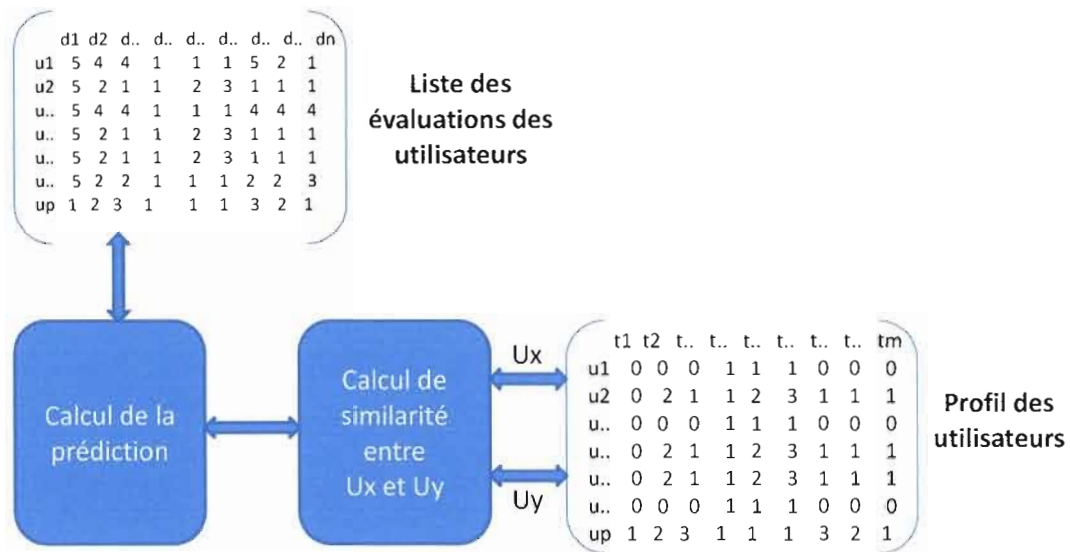


Figure 6.3 Filtrage hybride

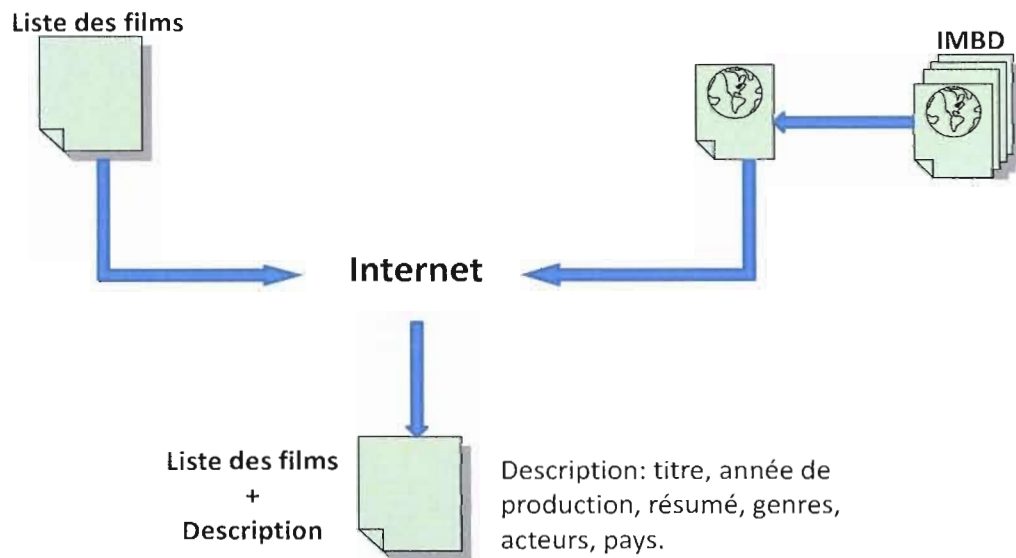


Figure 6.4 Prétraitement : Téléchargement du contenu

Le reste des étapes sont semblables à l'approche précédente de filtrage collaboratif. La seule différence est que nous utilisons la matrice *UserVSTerm*, au lieu de la matrice *UserVSItem*, pour le calcul de la similarité entre utilisateurs.

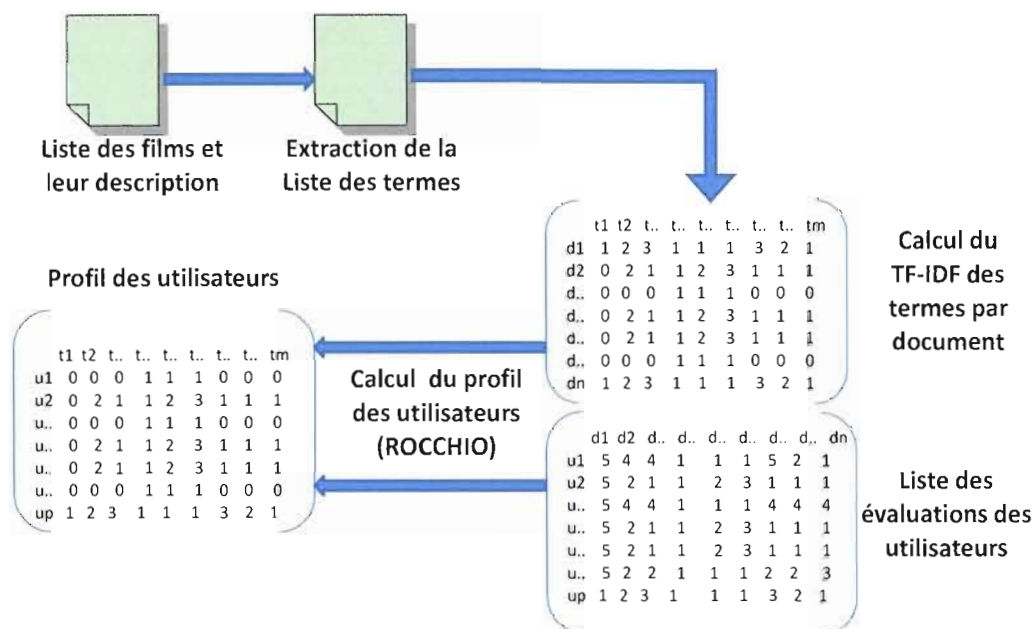


Figure 6.5 Prétraitement : Création du profil du document et de l'utilisateur

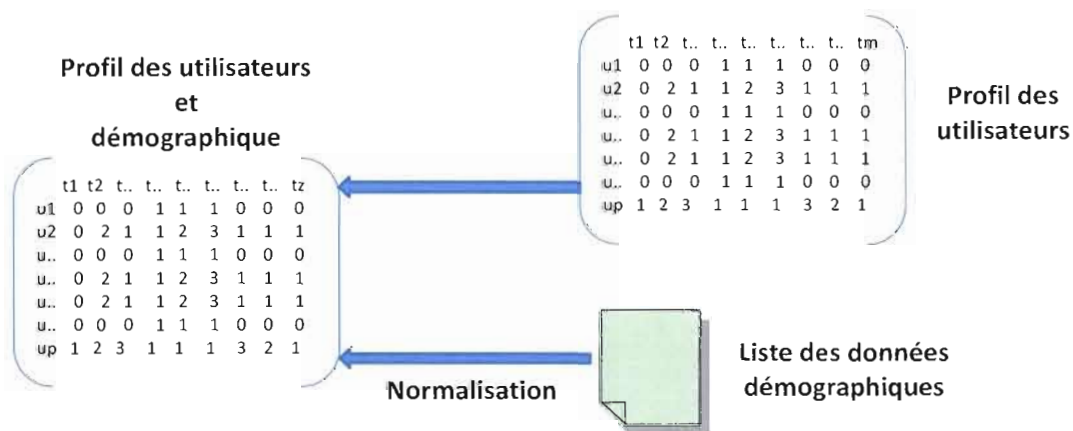


Figure 6.6 Prétraitement : Création du profil utilisateur démographique

Approche hybride démographique. Pour cette approche, les étapes sont semblables à celles de l'approche de filtrage hybride. La seule différence est que nous ajoutons au profil d'utilisateur (UserVSTerm) quelques informations démographiques normalisées (UserVSDem) sur les utilisateurs (âge, genre, métier, code postal). Cette étape est illustrée par la figure 6.6. Les étapes, de calcul des prédictions, sont présentées en détail dans la figure 6.7.

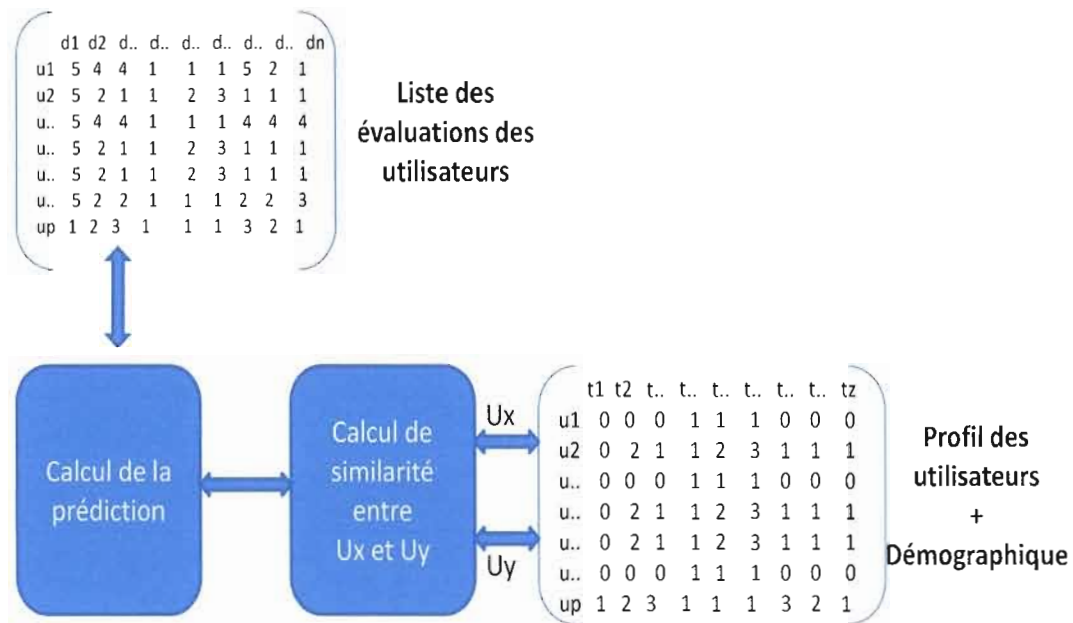


Figure 6.7 Filtrage hybride démographique

6.2.4. Protocole d'évaluation

Nous adoptons un procédé expérimental préalablement validé (Miller, Konstan et Riedl, 2004 ; Zaier, Godin et Faucher, 2008b). De récentes études montrent l'impact de la taille du voisinage sur la performance des différents algorithmes de recommandation (Zaier, Godin et Faucher, 2007). Ainsi, pour nos expérimentations, des ensembles d'apprentissage de taille variable allant de 50 à 943 utilisateurs ont été constitués en choisissant les premiers utilisateurs. L'ensemble de tests, quant à lui, demeure toujours le même.

Dans une première expérimentation évaluant les approches de filtrage collaboratif, filtrage hybride, et filtrage hybride démographique, nous avons procédé à des expériences où, successivement, nous avons commencé par un cycle d'entraînement sur une base d'apprentissage pour chacune des approches ; suivies par un cycle de test, sur la base d'évaluation, dans lequel chaque approche de filtrage produit d'un ensemble de prédictions ; et pour finir, nous avons appliqué nos métriques, présentées dans la section 6.2.2, à cet ensemble de prédictions et nous avons collecté les résultats.

Après cette opération, et jusqu'à la fin de la base d'apprentissage (943 utilisateurs), 50 nouveaux voisins sont ajoutés à cette dernière et un autre cycle entraînement/test est exécuté. Cette dernière étape simule la recherche des voisins dans un système distribué.

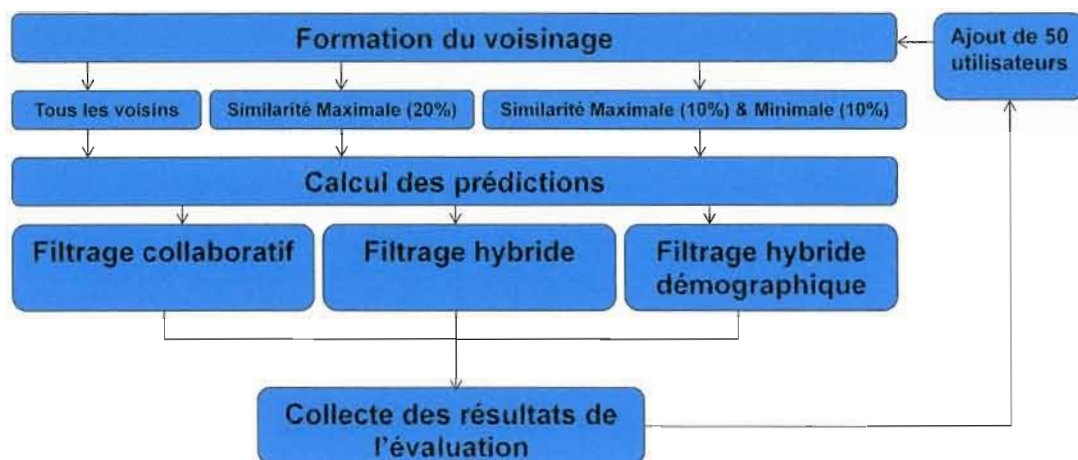


Figure 6.8 Protocole d'évaluation

Pour la deuxième expérimentation, basée sur le même protocole de la première, nous choisissons, pour chaque cycle d'entraînement/test, 20 % des utilisateurs avec la similarité maximale. Enfin, pour la dernière expérimentation, comme pour la deuxième, nous choisissons un sous-ensemble de voisins, mais avec 10 % des utilisateurs avec la similarité la plus forte et 10 % des utilisateurs avec la similarité la plus faible (les extrêmes). La figure 6.8 illustre les différentes étapes du protocole d'évaluation.

6.3. Résultats et discussions

Chacune des 7 collections, introduites dans la section 6.2.1, donne des résultats équivalents. Cependant, les résultats présentés dans cette section sont ceux obtenus avec le corpus *UI* de *MovieLens*. Nous comparons maintenant chacune des méthodes de recommandation, présentées dans la section 6.2.3, en fonction des métriques définies dans la section 6.2.2 et du protocole d'évaluation introduit dans la section 6.2.4.

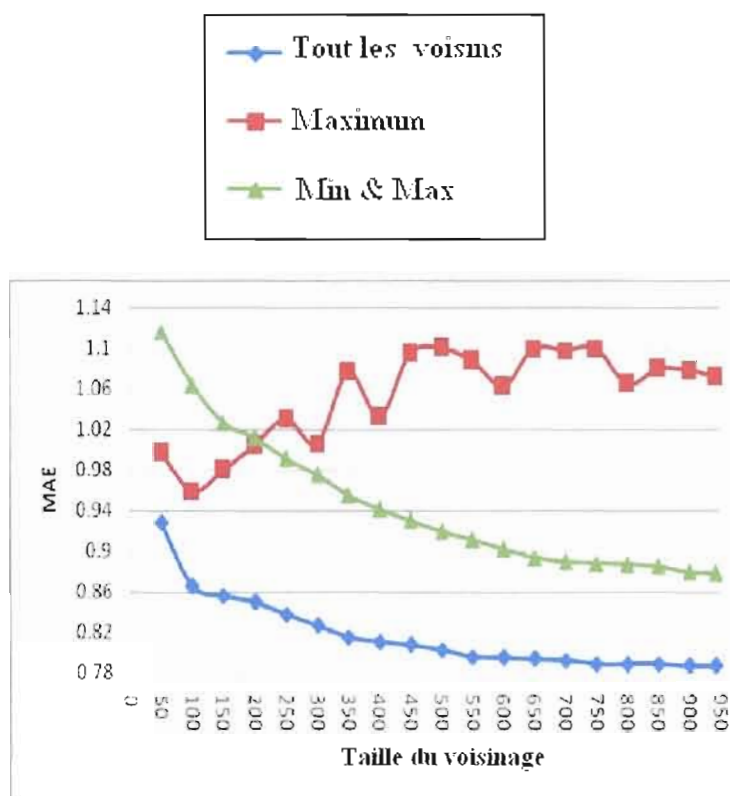


Figure 6.9 MAE pour le filtrage collaboratif en fonction de la taille du voisinage

Erreur absolue moyenne. Les figures 6.9 à 6.11 présentent la précision des prédictions fournies par les trois approches, à savoir le filtrage collaboratif, le filtrage hybride et le filtrage démographique. Nous notons une modeste différence entre les trois architectures (tous les voisins, le maximum et les extrêmes) en ce qui concerne les approches de filtrage hybride et de filtrage démographique. Cependant, l'approche de filtrage collaboratif démontre un avantage considérable pour l'architecture de tous les voisins. De plus, elle prouve que pour le filtrage hybride et le filtrage démographique l'architecture, basée sur la sélection des similarités maximales, est raisonnablement efficace.

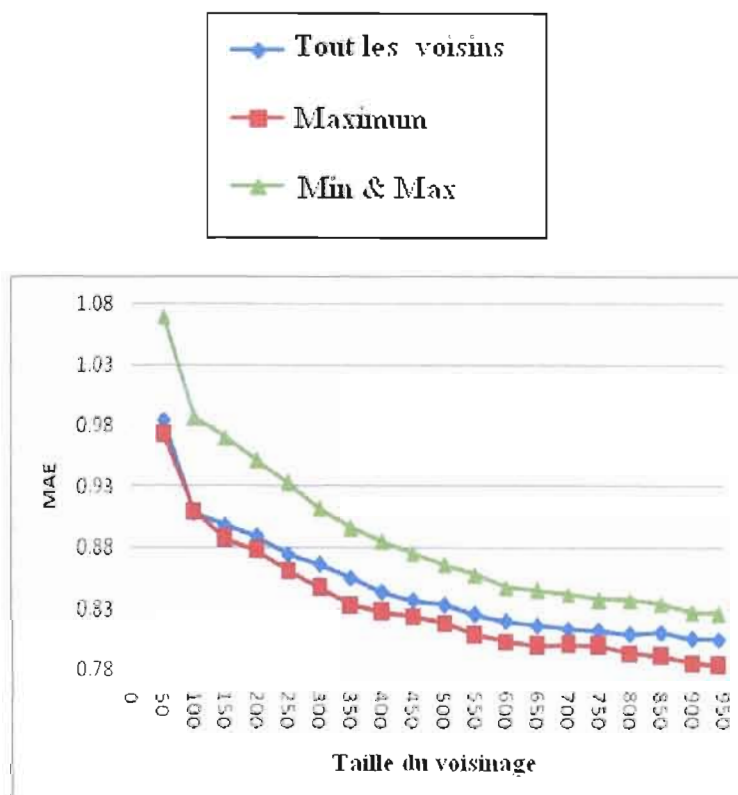


Figure 6.10 MAE pour le filtrage hybride en fonction de la taille du voisinage

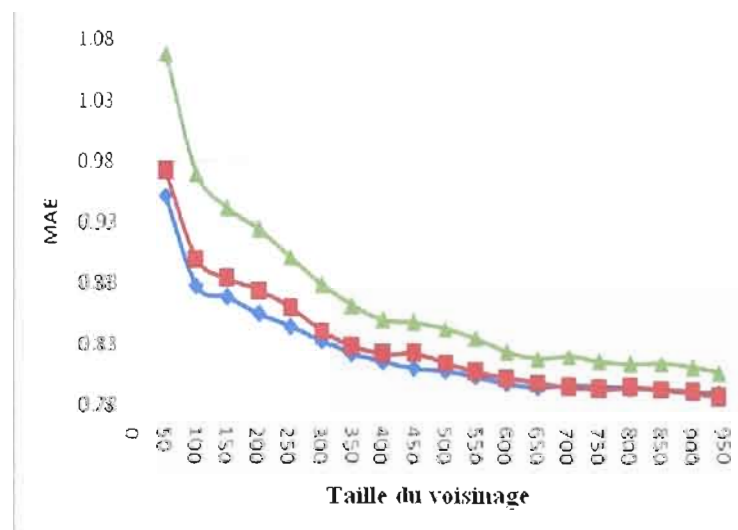


Figure 6.11 MAE pour le filtrage hybride démographique en fonction de la taille du voisinage

Rappel. Les figures 6.12 à 6.14 montrent la qualité de recommandation pour les trois approches de filtrage. Nous observons que, pour les trois architectures de discrimination, les approches de filtrage hybride et de filtrage démographiques sont équivalentes. Cependant, l'approche de filtrage collaboratif est la première à atteindre la valeur optimale pour les trois architectures (tous les voisins, le maximum et les extrêmes). De même, nous notons que, comme pour l'erreur absolue moyenne, l'approche de filtrage collaboratif montre un avantage substantiel pour l'architecture de tous les voisins.

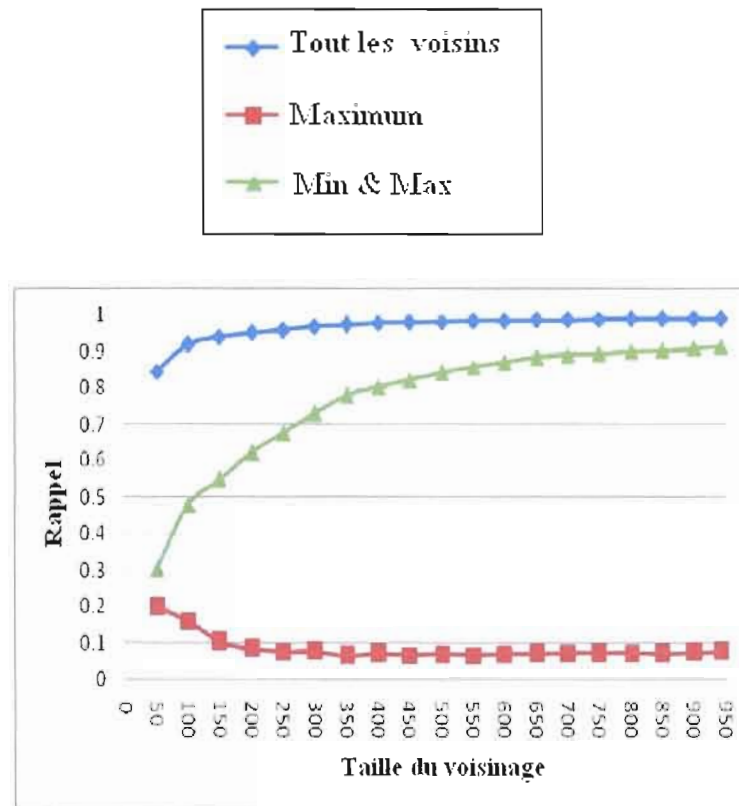


Figure 6.12 Rappel pour le filtrage collaboratif en fonction de la taille du voisinage

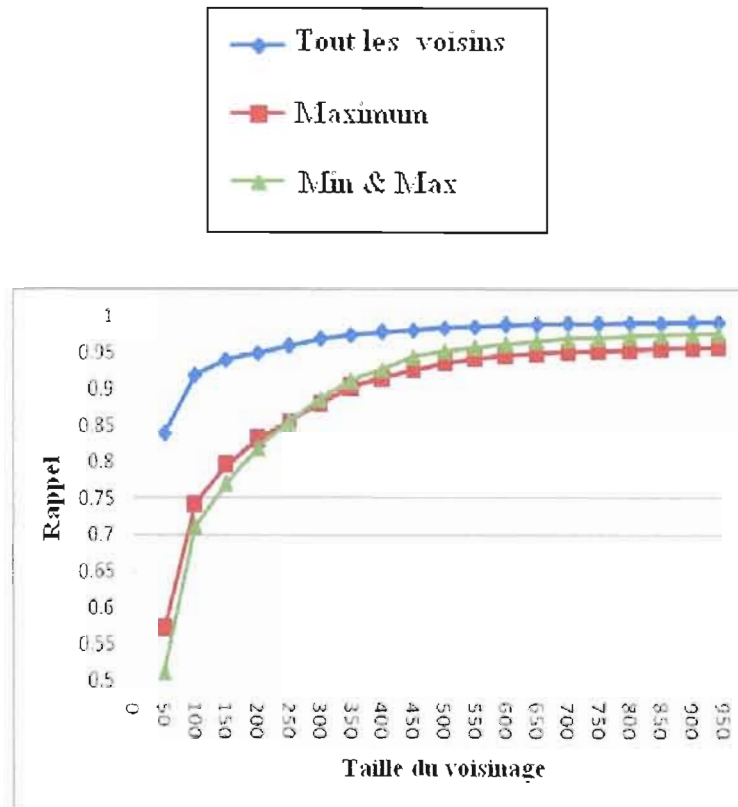


Figure 6.13 Rappel pour le filtrage hybride en fonction de la taille du voisinage

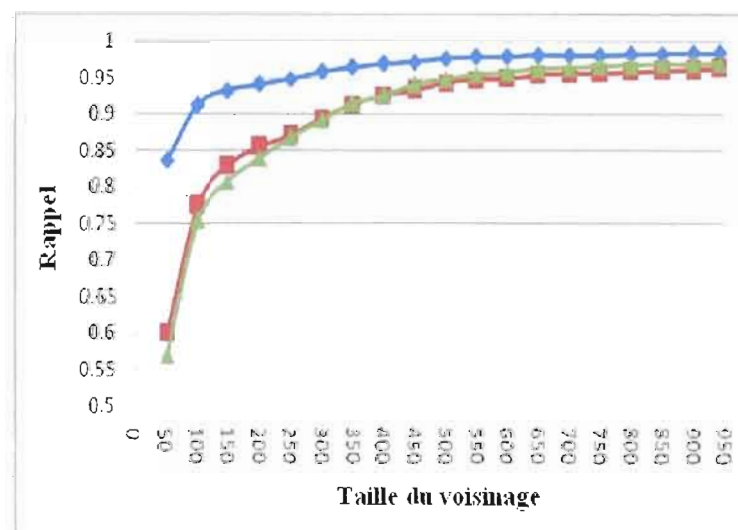


Figure 6.14 Rappel pour le filtrage hybride démographique en fonction de la taille du voisinage

Couverture. Les courbes représentées par les figures 6.15 à 6.17 présentent la qualité du voisinage. Ces figures, comme pour le rappel, indiquent que la couverture est fortement équivalente entre les approches de filtrage hybride et de filtrage démographique. De plus, nous notons que l'approche de filtrage démographique atteint la valeur optimale avant le filtrage hybride. En outre, nous observons que, comme pour l'erreur absolue moyenne et le rappel, l'approche de filtrage collaboratif montre un avantage consistant pour l'architecture de tous les voisins. Enfin, nous notons que les courbes sur les figures 6.12 à 6.14 et les figures 6.15 à 6.17 sont fortement similaires. Ceci peut être expliqué par le fait que ces deux métriques évaluent la qualité du voisinage.

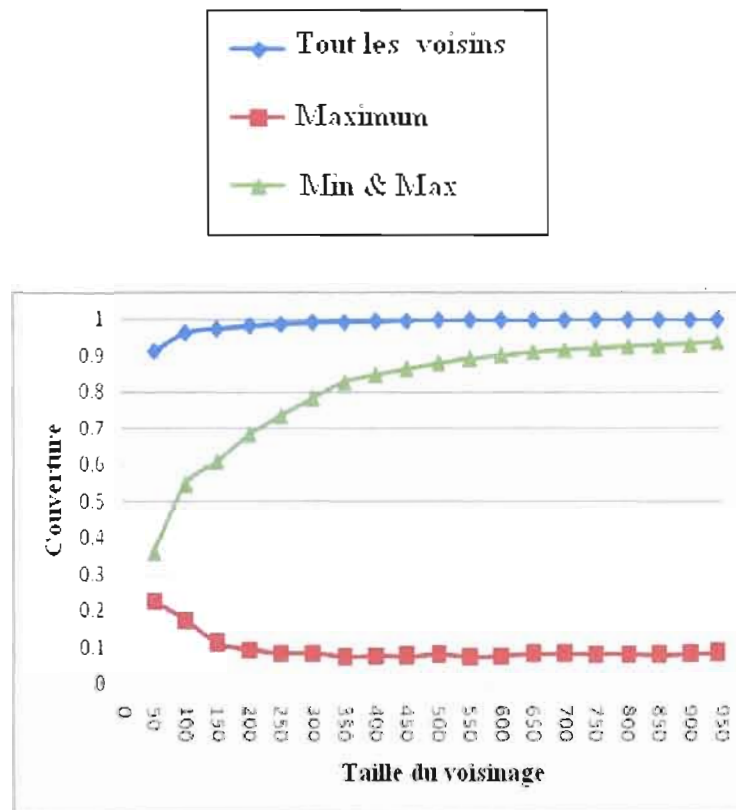


Figure 6.15 Couverture pour le filtrage collaboratif en fonction de la taille du voisinage

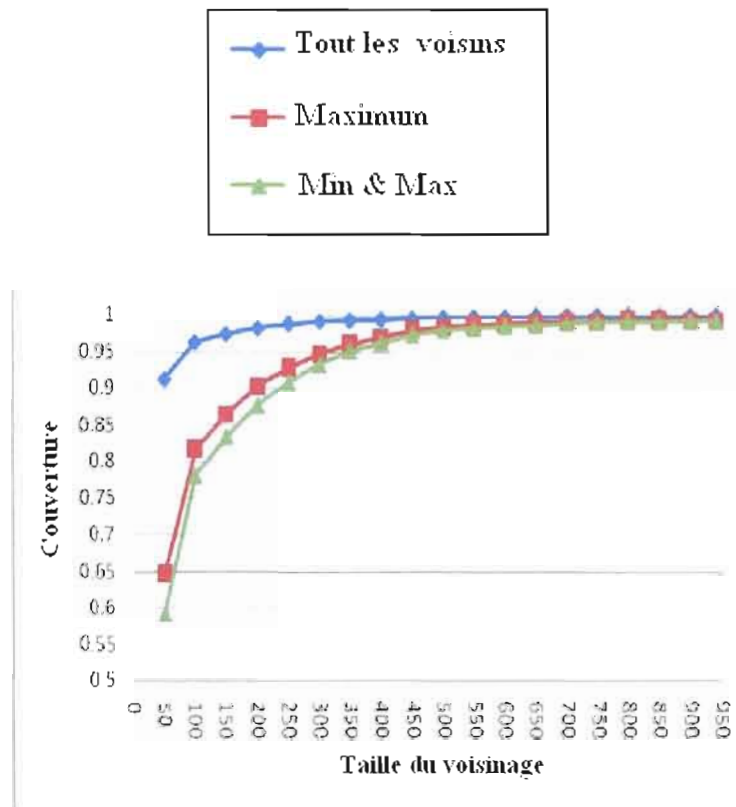


Figure 6.16 Couverture pour le filtrage hybride en fonction de la taille du voisinage

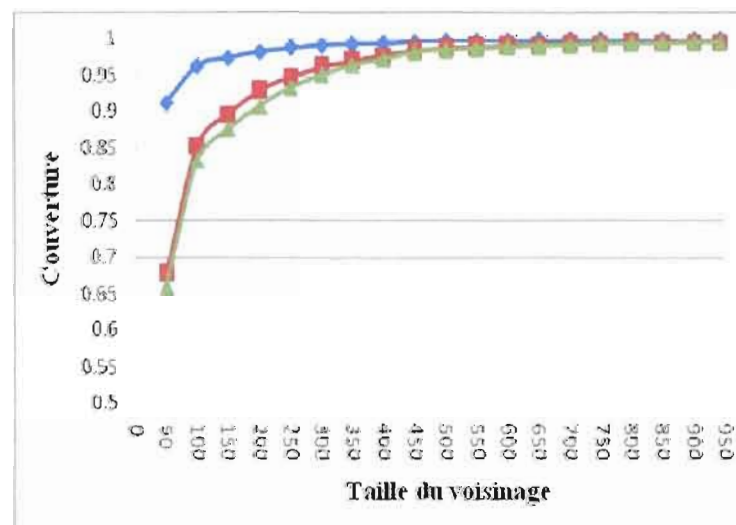


Figure 6.17 Couverture pour le filtrage hybride démographique en fonction de la taille du voisinage

Pour conclure, en vue des résultats fournis par les métriques ci-haut, nous pouvons affirmer que la qualité des recommandations fournies par notre approche est assez bonne. En effet, l'erreur absolue moyenne est fortement équivalente entre les différentes architectures. Par ailleurs, pour la couverture et le rappel, notre architecture basée sur les extrêmes atteint la valeur optimale légèrement après l'architecture de tous les voisins. Cependant, la complexité computationnelle de notre architecture est largement inférieure.

Chapitre VII

Conclusion et perspectives

« Ce n'est pas la fin. Ce n'est même pas le commencement de la fin. Mais, c'est peut-être la fin du commencement. »

– Winston Churchill (1874 – 1965)

7.1. Résumé

Les systèmes de recommandation deviennent, indubitablement, de plus en plus populaires. Cette popularité est le fruit de l'assistance fournie par ces outils aux utilisateurs dans leurs recherches, les rendant à la fois plus personnalisées et plus optimales. De plus, les travaux de recherches ne se limitent pas seulement au commerce électronique, mais sont actuellement sollicités dans plusieurs autres domaines d'application (Zaier, Godin et Faucher, 2008a). Par exemple, *Abyss eurocom* utilise les systèmes de recommandation comme guide dans des musées virtuels (Berrut et Denos, 2003); *iKarma* et *eBay* les exploitent pour permettre aux internautes de se créer des réputations (Stoti, 2006); *RAZOR2* et *CASSANDRA* les utilisent pour le filtrage des pourriels (Gray et Haahr, 2004 ; Kong *et al.*, 2006 ; Prakash et O'Donnell, 2005); *NewsGator* et *Illumio* les exploitent pour recommander des billets de différents blogues (Zaier, 2008a); *ColFi* les utilise pour les services de rencontre (Brozovsky, 2007); *Lifestyle* les exploite dans les interactions de l'avatar avec l'utilisateur (Ujjiin et Bentley, 2001); Andre Vellino a mis en évidence l'apport possible d'un système de recommandation pour une bibliothèque scientifique (Vellino et Zeber, 2008).

Cependant, tous ces systèmes présentent quelques limitations qui ont été mises en évidence dans le premier chapitre de cette thèse. Ainsi, notre recherche a pris naissance dans le but de pallier les contraintes des systèmes de recommandation existants. À cet effet, nous

nous sommes intéressés à quelques pistes de recherche pour répondre à cette problématique. En effet, lors de notre revue de la littérature, nous avons pu identifier les travaux les plus pertinents pour notre sujet de recherche. Ainsi, cette thèse tourne autour de 3 sujets interdisciplinaires, à savoir : le filtrage l'information, les architectures distribuées et les systèmes multi-agents, introduits en détail dans les chapitres précédents (Adomavicius et Tuzhilin, 2005) :

Système de recommandation. Ce système utilise les opinions et les comportements d'une communauté d'utilisateurs afin de les guider, parmi un ensemble considérable de choix, vers du contenu susceptible de les intéresser. La motivation des systèmes de recommandation est d'étendre la notion de bouche à oreille entre amis à des milliers d'internautes. De plus, avec les systèmes de recommandation, l'internaute ne recherche plus l'information, l'information vient directement à lui (Liu et Shih, 2005).

Architecture décentralisée. La méthode décentralisée utilise chaque internaute comme un pseudo-serveur et donc, ne possède aucun serveur centralisé. Cette méthode a l'avantage d'être plus robuste en cas d'interruption de service et plus mobile (Miller, Konstan et Riedl, 2004 ; Ziegler, 2005).

Système multi-agents. Un système multi-agents est constitué d'un ensemble de processus informatiques se déroulant en même temps, donc de plusieurs agents vivant au même moment, partageant des ressources communes et communiquant entre eux. Le point clé des systèmes multi-agents, pour les systèmes de recommandation, réside dans les interactions sociales entre les agents (Gómez-Sanz, Pavón et Díaz-Carrasco, 2003).

Ces 3 sujets principaux ont été associés pour créer un système de recommandation décentralisé. Ce modèle, présenté en détail dans le chapitre 5, nous a permis de surmonter une grande partie des lacunes des systèmes de recommandation introduites dans le premier chapitre. Néanmoins, il est à noter que notre solution est une des solutions possibles pour remédier aux lacunes des systèmes de recommandations classiques. En plus, malgré que notre modèle se limite à la recommandation de film, ce dernier peut être utilisé, avec seulement quelques ajustements mineurs, dans n'importe quel domaine évoqué plus haut.

7.2. Discussion et perspective

Compte tenu de l'état actuel de notre recherche et du domaine pluridisciplinaire de celle-ci, les contributions scientifiques de la réalisation du projet proposé sont au nombre de six. 1) l'identification des problèmes spécifiques aux systèmes courants de filtrage d'information ; 2) des avancées significatives dans la mise en œuvre de profils adaptatifs d'utilisateurs, qui mettent à profit les informations sur l'article et démographiques de l'utilisateur ; 3) une mise au point de méthodes d'expérimentation et d'évaluation ; 4) une étude sur les modèles sociocognitifs dans les systèmes multi-agents ; 5) la mise en œuvre d'une technologie « *Push* » souple et adaptée et qui permet un meilleur recouvrement de l'espace d'information ; 6) une architecture de recommandation distribuée.

Cependant, la recherche présentée ici en est à ses débuts. En effet, nos futurs travaux prendront plusieurs directions. Nous pouvons classer ces derniers en deux catégories. D'une part, les perspectives à court terme dans lesquelles nous prévoyons réaliser des expérimentations plus exhaustives. Nous comptons réaliser des expérimentations avec un grand nombre d'utilisateurs et dans des situations de filtrage d'informations réelles. Ainsi, nous pouvons identifier et mesurer les différents comportements émergeant des interactions entre agents. En effet, étant donné que l'aspect comportements sociaux n'est pas le point central de cette recherche, nous nous sommes limités, dans l'analyse empirique, à l'évaluation de l'approche proposée pour notre système de recommandation. Nous pensons cependant qu'une telle analyse pourrait donner des résultats intéressants. En plus, nous prévoyons identifier d'autres métriques d'évaluation capable d'évaluer en même temps la qualité de la prédiction et la pertinence des documents (Adomavicius et Tuzhilin, 2005). En outre, nous allons essayer d'étudier l'impact des différentes architectures pair-à-pair sur la performance de notre système de recommandation (Miller, Konstan et Riedl, 2004). Dans le même ordre d'idée, nous envisageons d'examiner l'effet de l'utilisation de différents algorithmes de pondération plus complexe sur la performance de notre système de recommandation. Étant donné que la pondération n'est pas le point central de cette recherche, nous nous sommes limités, dans cette dernière, à l'utilisation d'un système de pondération simple. Cependant, il existe, dans la littérature, plusieurs autres algorithmes de

pondération susceptible d'améliorer la qualité des recommandations de notre système (Amati et Joost Van Rijsbergen, 2002 ; Manning, Raghavan et Schtze, 2008). Enfin, nous allons essayer, comme Amazon, d'expliquer les recommandations faites par le système pour qu'elle soit mieux acceptée par l'utilisateur et pour améliorer les recommandations futures (Tintarev et Masthoff, 2007). Pour ce faire, nous pouvons utiliser, une explication textuelle ou graphique (Vellino et Zeber, 2008).

D'autre part, en ce qui concerne nos perspectives à long terme, nous devons prendre en considération la protection des renseignements privés constituant le profil utilisateur (information personnelle, intérêts, goûts, habitudes, etc.). Le fait que notre approche se base sur une architecture distribuée, où les données sont stockées en local, assure une protection pour ce type d'information. Cependant, vu la nature de l'information, ces systèmes doivent assurer une protection de la vie privée plus avancée. Pour expliquer la difficulté de ce type de protection dans un environnement collaboratif, nous pouvons donner l'exemple du problème des deux millionnaires. Supposons qu'Alice et Bob sont deux millionnaires qui veulent découvrir qui d'entre eux est le plus riche sans rendre public le montant précis de leur fortune respective (Zaier, 2008b). Dans ce but, nous pouvons penser à l'utilisation des surnoms pour garder l'anonymat et de la cryptographie pour le transfert de données. Dans le même ordre d'idées, nous devons penser à un système d'identification unique de l'utilisateur. En effet, la notion de confiance ne peut pas être utilisée si un utilisateur peut changer d'identification toutes les fois qu'il le souhaite. Finalement, nous pouvons aussi utiliser des techniques, empruntées au domaine de la fouille de données (*Datamining*), de perturbation ou d'obscurcissement (*Obfuscation*) des données (Aimeur *et al.*, 2006 ; Aimeur *et al.*, 2008 ; Kobsa, 2007 ; Lam, Frankowski et Riedl, 2006 ; Lam et Riedl, 2005 ; Lathia, Hailes et Capra, 2007). Cependant, les techniques de protection de la vie privée nuisent à la performance des systèmes de recommandation. Il faut donc trouver un compromis entre la sécurité et la performance du système (Canny, 2002b ; Canny, 2002a ; Olsson, 2003).

Tandis que notre recherche, comme la plupart des systèmes courants de recommandation, se base sur les systèmes de recommandation utilisant un simple critère d'évaluation, beaucoup d'applications commerciales ont commencé à employer des méthodes d'évaluation multicritères. *Yahoo! Cinéma* demande à ses utilisateurs de fournir quatre critères

spécifiques au sujet du film : histoire, interprétation, réalisation, et images (<http://fr.movies.yahoo.com>) ; *Zagat* fournit trois critères pour évaluer un restaurant : nourriture, décor, et service (<http://www.zagat.com>) ; *Circuitcity* propose quatre critères pour évaluer un article : l'affichage, la performance, la durée de vie de la batterie, et le coût (<http://www.circuitcity.com>). L'évaluation multicritères est considérée comme une valeur ajoutée aux systèmes courants de recommandation (Adomavicius et YoungOk, 2007). Dans nos futurs travaux, nous étudierons les techniques de recommandation qui profitent de l'évaluation multicritères pour améliorer la qualité de recommandation. Plus précisément, nous allons essayer d'adapter notre modèle de recommandation à un système de recommandation multicritère.

Un autre point intéressant est l'intégration des comportements utilisateur dans le calcul du degré d'intérêt d'un article (Chan, 1999 ; Claypool *et al.*, 2001). En effet, nous pouvons utiliser les comportements produits par l'utilisateur, en consultant un document (imprimer, défiler, mettre en favori, enregistrer, etc.) ou en écoutant sa musique (sauvegarder, nombre de fois écouté, dernière fois écoutée, etc.), pour inférer les préférences des utilisateurs (Chan, 1999 ; Claypool *et al.*, 2001 ; Miller, 2003 ; Ziegler, 2005). Enfin, un autre grand point, c'est le partage de la puissance de calcul. En effet, nous pouvons imaginer que si un agent ne possède pas assez de ressources (mémoires, processeur, etc.), il peut demander des recommandations à un autre agent dans lequel il a confiance. Ce concept, appelé délégation, a été introduit par Castelfranchi (1998) comme étant un aspect primordial de la coopération des agents. Cependant, selon Castelfranchi (1998), la délégation implique une confiance absolue dans l'autre agent des points de vue de la fiabilité et de la crédibilité (Lorini *et al.*, 2007).

En conclusion, il est évident que les systèmes de recommandation joueront un rôle de plus en plus significatif dans le futur. Cet intérêt croissant ouvre logiquement sur des perspectives très larges et sur plusieurs directions de recherche. Cette thèse a abordé plusieurs de ces directions et a proposé une nouvelle approche de système de recommandation à la fois plus social et plus mobile (distribué). Il reste cependant de nombreuses pistes de recherche à explorer. Le sujet demeure donc très ouvert, et il est bien

évident que plusieurs de ces voies sont extrêmement intéressantes et seront à suivre dans un futur proche.

BIBLIOGRAPHIE

- Adda , Mehdi, Petko Valtchev, Rokia Missaoui et Chabane Djeraba. 2007. «An Ontology-based sequential pattern mining approach for recommendation purposes ». *IEEE Internet Computing*. vol. 11, no 4, p. 45-52.
- Adomavicius, Gediminas, et Alexander Tuzhilin. 2005. «Toward the next generation of recommender systems: a survey of the state-of-the-art and possible extensions». *Transactions on Knowledge and Data Engineering*. vol. 17, no 6, p. 734-749.
- Adomavicius, Gediminas, et Kwon YoungOk. 2007. «New Recommendation Techniques for Multicriteria Rating Systems». *Intelligent Systems*. vol. 22, no 3, p. 48-55.
- Aggarwal, Charu C., Joel L. Wolf, Kun-Lung Wu et Philip S. Yu. 1999. «Horting Hatches an Egg: A New Graph-Theoretic Approach to Collaborative Filtering». *Knowledge Discovery and Data Mining*, p. 201-212.
- Agre, E. Philip, et David Chapman. 1995. «Pengi: an implementation of a theory of activity». In *Computation & intelligence: collected readings*, p. 635-644: American Association for Artificial Intelligence.
- Aimeur, E., G. Brassard, J. M. Fernandez et F. S. Mani Onana (2006). Privacy-preserving demographic filtering. SAC '06: Proceedings of the 2006 ACM symposium on Applied computing. Dijon, France, ACM: 872-878 p En ligne. <<http://doi.acm.org/10.1145/1141277.1141479>>.
- Aimeur, Esma, Gilles Brassard, Jose M. Fernandez, Flavien Serge Mani Onana et Zbigniew Rakowski. 2008. «Experimental Demonstration of a Hybrid Privacy-Preserving Recommender System». *ares*. vol. 0, p. 161-170. En ligne. <<http://doi.ieeecomputersociety.org/10.1109/ARES.2008.193>>.

- Amati, Gianni , et Cornelis Joost Van Rijsbergen. 2002. «Probabilistic models of information retrieval based on measuring the divergence from randomness». *ACM Trans. Inf. Syst.* vol. 20, no 4, p. 357-389.
- Anderson, Chris. 2004. «The Long Tail». *Wired Magazine*. vol. 12, no 10. En ligne. <<http://www.wired.com/wired/archive/12.10/tail.html>>.
- , 2006. *The Long Tail: Why the Future of Business Is Selling Less of More*: Hyperion.
- Androutsellis-Theotokis, Stephanos, et Diomidis Spinellis. 2004. «A survey of peer-to-peer content distribution technologies». *ACM Comput. Surv.* vol. 36, no 4, p. 335-371. En ligne. <<http://doi.acm.org/10.1145/1041680.1041681>>.
- Ansari, A., S. Essegiaier et R. Kohli. 2000. «Internet recommendations systems». *Journal of Marketing Research*. vol. 37, no 3, p. 363-375.
- Balabanovi, Marko, et Yoav Shoham. 1997. «Fab: content-based, collaborative recommendation». *Commun. ACM*. vol. 40, no 3, p. 66-72.
- Basilico, Justin, et Thomas Hofmann. 2004. «Unifying collaborative and content-based filtering». *ICML '04: Proceedings of the twenty-first international conference on Machine learning*, p. 9. En ligne. <<http://doi.acm.org/10.1145/1015330.1015394>>.
- Basu, Chumki, Haym Hirsh et William W. Cohen. 1998. «Recommendation as Classification: Using Social and Content-Based Information in Recommendation». *AAAI/IAAI*, p. 714-720.

- Belkin, Nicholas J., et W. Bruce Croft. 1992. «Information filtering and information retrieval: two sides of the same coin?». *Commun. ACM*. vol. 35, no 12, p. 29-38. En ligne. <<http://doi.acm.org/10.1145/138859.138861>>.
- Bell, Robert, et Yehuda Koren. 2007a. «Improved Neighborhood-based Collaborative Filtering».
- . 2007b. «Lessons from the Netflix prize challenge». *SIGKDD Explor. Newsl.* vol. 9, no 2, p. 75-79.
- . 2007c. «Scalable Collaborative Filtering with Jointly Derived Neighborhood Interpolation Weights». In *Data Mining, 2007. ICDM 2007. Seventh IEEE International Conference on*.
- Bell, Robert, Yehuda Koren et Chris Volinsky. 2007a. «The BellKor solution to the Netflix Prize».
- . 2007b. «Chasing \$1,000,000: How We Won The Netflix Progress Prize ». *Statistical Computing and Statistical Graphics Newsletter*. vol. 18, p. 4-12.
- . 2007c. «Modeling relationships at multiple scales to improve accuracy of large recommender systems». In *Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining*: ACM.
- Berrut, Catherine, et Nathalie Denos. 2003. «Filtrage collaboratif». In *Assistance intelligente à la recherche d'informations*, Hermes Science Publications, p. 30. En ligne. <<http://www-mrim.imag.fr/publications/2003/CB001/berrut03b.pdf>>.
- Billsus, Daniel, et Michael J. Pazzani. 1998. «Learning collaborative information filters». *Proc. 15th International Conf. on Machine Learning*, p. 46-54.

- , 1999. «A personal news agent that talks, learns and explains». *The third annual conference on Autonomous Agents (AGENTS '99)*, p. 268-275. En ligne. <<http://doi.acm.org/10.1145/301136.301208>>.
- , 2000. «User Modeling for Adaptive News Access». *User Modeling and User-Adapted Interaction*. vol. 10, no 2-3, p. 147-180.
- Boesch, C., et H. Boesch. 1989. «Hunting behaviour of wild chimpanzees in the Tai National Park, Ivory Coast». *American Journal of Physical Anthropology*. vol. 78, no 4, p. 547-574.
- Bonabeau, Eric, et Guy Theraulaz. 1994. *Intelligence Collective*: Hermes.
- Breese, John S., David Heckerman et Carl Kadie. 1998. «Empirical Analysis of Predictive Algorithms for Collaborative Filtering». *Proceedings of the Fourteenth Conference on Uncertainty in Artificial Intelligence (UAI-98)*, p. 43-52.
- Brozovsky, Lukas. 2007. «ColFi - Recommender System for a Dating Service». In *ZNALOSTI 2007 conference*.
- Burke, Robin. 2002. «Hybrid Recommender Systems: Survey and Experiments». *User Modeling and User-Adapted Interaction*. vol. 12, no 4, p. 331-370. En ligne. <<http://dx.doi.org/10.1023/A:1021240730564>>.
- Burke, Robin, Bamshad Mobasher, Chad Williams et Runa Bhaumik (2006). Classification features for attack detection in collaborative recommender systems. KDD '06: Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining. Philadelphia, PA, USA, ACM: 542-547 p En ligne. <<http://doi.acm.org/10.1145/1150402.1150465>>.

- Byrne, Richard W. 1995. *The thinking ape evolutionary origins of intelligence*. Oxford: Oxford University Press.
- , 2006. «Parsing Behavior: A Mundane Origin for an Extraordinary Ability?». In *Roots of Human Sociality. Culture, Cognition and Interaction*, Nicholas J. Enfield et Stephen C. Levinson, p. 478-505. Oxford: Berg.
- Byrne, Richard W., et Lucy A. Bates. 2007. «Sociality, Evolution and Cognition». *Current Biology*. vol. 17, no 16, p. R714-R723. En ligne. <<http://www.sciencedirect.com/science/article/B6VRT-4PG33JY-10/1/3a41798a6800bf4cc151f5ec6dd2e222>>.
- Byrne, Richard W., et Andrew Whiten. 1988. *Machiavellian intelligence social expertise and the evolution of intellect in monkeys, apes and humans*, Pbk. Oxford: Clarendon Press.
- Canny, John. 2002a. «Collaborative filtering with privacy via factor analysis». In *Proceedings of the 25th annual international ACM SIGIR conference on Research and development in information retrieval*: ACM.
- Canny, John 2002b. «Collaborative Filtering with Privacy». In *Proceedings of the 2002 IEEE Symposium on Security and Privacy*: IEEE Computer Society.
- Castelfranchi, C. 1990. «Social Power: A Point Missed in Multi-Agent, DAI and HCI». *Decentralized A.I. : Proc1 of the First European Workshop on Modelling Autonomous Agents in a Multi-Agent World, Cambridge, England (Amsterdam)*, p. 49-62.
- Castelfranchi, Cristiano. 1998. «Modelling social action for AI agents». *Artif. Intell.* vol. 103, no 1-2, p. 157-182.

- , 2000. «Artificial liars: Why computers will (necessarily) deceive us and each other». *Ethics and Inf. Technol.* vol. 2, no 2, p. 113-119.
- Chan, Philip K. 1999. «A non-invasive learning approach to building web user profiles». *Workshop on Web usage analysis and user profiling, Fifth International Conference on Knowledge Discovery and Data Mining*.
- Cherniack, Mitch, Eduardo F. Galvez, Michael J. Franklin et Stan Zdonik. 2003. «Profile-Driven Cache Management». *Data Engineering, International Conference on*. vol. 0, p. 645. En ligne. <<http://doi.ieeecomputersociety.org/10.1109/ICDE.2003.1260828>>.
- Chien, Y. H., et E. George. 1999. «A Bayesian model for collaborative filtering». In *the Seventh International Workshop on Artificial Intelligence and Statistics*.
- Clarke, Ian, Oskar Sandberg, Brandon Wiley et Theodore W. Hong. 2000. «Freenet: A Distributed Anonymous Information Storage and Retrieval System». *Workshop on Design Issues in Anonymity and Unobservability*, p. 46-66.
- Claypool, M., M. Claypool, D. Brown, P. Le et M. A. Waseda M. Waseda. 2001. «Inferring user interest
- Inferring user interest». *Internet Computing, IEEE*. vol. 5, no 6, p. 32-39.
- Claypool, M., A. Gokhale, T. Miranda, P. Murnikov, D. Netes et M. Sartin. 1999. «Combining Content-Based and Collaborative Filters in an Online Newspaper». In *SIGIR '99 Workshop Recommender Systems: Algorithms and Evaluation*: ACM.
- Computer Industry Almanac. 2006. «Worldwide Internet Users Top 1 Billion in 2005». Computer Industry Almanac. En ligne. <www.c-i-a.com/pr0106.htm>.

- Condliff, Michelle Keim, David D. Lewis et David Madigan. 1999. «Bayesian Mixed-Effects Models for Recommender Systems». In *SIGIR '99 Workshop on Recommender Systems: Algorithms and Evaluation*. En ligne. <<http://citeseer.ist.psu.edu/234817.html>>.
- Connor, R. C. 2007. «Dolphin social intelligence: complex alliance relationships in bottlenose dolphins and a consideration of selective environments for extreme brain size evolution in mammals». *Philos Trans R Soc Lond B Biol Sci*. vol. 362, no 1480, p. 587-602.
- Connor, Richard C., Rachel Smolker et Lars Bejder. 2006. «Synchrony, social behaviour and alliance affiliation in Indian Ocean bottlenose dolphins, *Tursiops aduncus*». *Animal Behaviour*. vol. 72, no 6, p. 1371-1378. En ligne. <<http://www.sciencedirect.com/science/article/B6W9W-4M3BCCF-2/1/28e60933e9636ee1d2fda069f3bdc5bb>>.
- Cosley, Dan, Shyong K. Lam, Istvan Albert, Joseph A. Konstan et John Riedl. 2003. «Is Seeing Believing? How Recommender Interfaces Affect Users' Opinions». In *CHI Lett.*
- Dasser, V. 1988. «A social concept in Java monkeys». *Animal Behaviour*. vol. 36, no 1, p. 225-230.
- De Waal, Frans. 2000. *Chimpanzee Politics: Power and Sex Among Apes*: Johns Hopkins University Press.
- Delgado, J. 1999. «Memory-Based Weighted Majority Prediction for Recommender Systems».
- Demazeau, Y., et J.-P. Müller. 1990. «Decentralized Artificial Intelligence». *Decentralized A.I. : Proc1 of the First European Workshop on Modelling Autonomous Agents in a Multi-Agent World, Cambridge, England (Amsterdam)*, p. 3-13.

- , 1991. «From Reactive to Intentional Agents». *Decentralized A.I. 2: Proc1 of the 2nd European Workshop on Modelling Autonomous Agents in a Multi-Agent World* (Amsterdam), p. 3-10.
- Deshpande, Mukund, et George Karypis. 2004. «Item-based top-N recommendation algorithms». *ACM Trans. Inf. Syst.* vol. 22, no 1, p. 143-177. En ligne. <<http://doi.acm.org/10.1145/963770.963776>>.
- Drogoul, A., et Jacques Ferber. 1994. «Multi-agent simulation as a tool for studying emergent processes in societies». In *Simulating Societies*.
- Drogoul, Alexis. 1993. «De la simulation multi-agents à la résolution collective de problèmes - Une étude de l'émergence de structures d'organisation dans les systèmes multi-agents». Paris, Université Paris VI.
- Drogoul, Alexis, et Anne Collinot. 1997. «Entre réductionnisme méthodologique et stratégie intentionnelle, l'éthologie, un modèle alternatif pour l'I.A.D. ?». In *JFIADSMA'97*: Hermès.
- Drogoul, Alexis, et C. Dubreuil. 1992. «Eco-Problem-Solving Model: Results of the N-Puzzle». *MAAMAW-91: Pre-Proc1 of the 3rd European Workshop on "Modeling Autonomous Agents and Multi-Agent Worlds"* (Kaiserslautern, Germany), p. 1-11.
- Drogoul, Alexis, et Jacques Ferber. 1993. «From Tom Thumb to the Dockers: some experiments with foraging robots». In *Proceedings of the second international conference on From animals to animats 2 : simulation of adaptive behavior: simulation of adaptive behavior*: MIT Press.
- Elberse, Anita (2008). Should You Invest in the Long Tail? Harvard Business Review En ligne. <http://harvardbusinessonline.hbsp.harvard.edu/hbsp/hbr/articles/article.jsp?ml_acti

on=get-
 article&articleID=R0807H&ml_issueid=BR0807&ml_subscriber=true&pageNumbe
 r=1&_requestid=113923>.

Ferber, Jacques. 2007. *Les Systèmes multi-agents: Vers une intelligence collective*: Dunod.

Foner, Leonard N. 1997. «Yenta: a multi-agent, referral-based matchmaking system». *AGENTS '97: Proceedings of the first international conference on Autonomous agents*, p. 301-307. En ligne. <10.1145/267658.267732>.

-----, 1999. «Political Artifacts and Personal Privacy: The Yenta Multi-Agent Distributed Matchmaking System». Boston, MA, USA, Massachusetts Institute of Technology.

Frankowski, Dan, Shyong K. Lam, Shilad Sen, F. Maxwell Harper, Scott Yilek, Michael Cassano et John Riedl. 2007. «Recommenders Everywhere: The WikiLens Community-Maintained Recommender System ». In *WikiSym '07: Proceedings of the 2007 International Symposium on Wikis*: ACM. En ligne. <<http://www.youtube.com/watch?v=bsgL5b9Dbz8>>.

Galliers, J. R. 1990. «The Positive Role of Conflict in Cooperative Multi-Agent Systems». *Decentralized A.I. : Proc1 of the First European Workshop on Modelling Autonomous Agents in a Multi-Agent World, Cambridge, England* (Amsterdam), p. 33-46.

Gantz, John F., David Reinsel, Christopher Chute, Wolfgang Schlichting, John McArthur, Stephen Minton, Irida Xheneti, Anna Toncheva et Alex Manfrediz. 2007. «The Expanding Digital Universe: A Forecast of Worldwide Information Growth Through 2010». IDC. En ligne. <http://www.emc.com/about/destination/digital_universe/pdf/Expanding_Digital_Universe_IDC_WhitePaper_022507.pdf>.

Gaul, Wolfgang, et Lars Schmidt-Thieme. 2002. «Recommender Systems Based On User Navigation Behaviour In The Internet». *Behaviormetrika*. vol. 29, no 1, p. 1-22.

- Gechter, F., et O. Simonin. 2005. «Conception de SMA réactifs pour la résolution de problèmes: Une approche basée sur l'environnement». In *12eme journées Francophones sur les Systèmes Multi-Agents*: Hermes Science.
- Getoor, L., et M. Sahami. 1999. «Using Probabilistic Relational Models for Collaborative Filtering». *Working Notes of the KDD Workshop on Web Usage Analysis and User Profiling*.
- Goldberg, Ken, Theresa Roeder, Dhruv Gupta et Chris Perkins. 2001. «Eigentaste: A Constant Time Collaborative Filtering Algorithm». *Information Retrieval*. vol. 4, no 2, p. 133-151.
- Gómez-Sanz, Jorge J., Juan Pavón et Áureo Díaz-Carrasco. 2003. «The PSI3 Agent Recommender System». *ICWE*. vol. 2722, p. 30-39.
- Good, Nathaniel, Ben J. Schafer, Joseph A. Konstan, Al Borchers, Badrul Sarwar, Jon Herlocker et John Riedl. 1999. «Combining collaborative filtering with personal agents for better recommendations». *AAAI '99/IAAI '99: Proceedings of the sixteenth national conference on Artificial intelligence and the eleventh Innovative applications of artificial intelligence conference innovative applications of artificial intelligence*, p. 439-446.
- Goth, Greg. 2006. «From the Editors: Peer-to-Peer Community-Looking beyond the Legacy of Napster and Gnutella». *IEEE Distributed Systems Online*. vol. 7, no 3, p. 5. En ligne. <<http://dx.doi.org/10.1109/MDSO.2006.20>>.
- Gray, Alan, et Mads Haahr. 2004. «Personalised, Collaborative Spam Filtering». En ligne. <<http://www.ceas.cc/papers-2004/132.pdf>>.

- Groh, Georg, et Christian Ehming. 2007. «Recommendations in taste related domains: collaborative filtering vs. social filtering». In *Proceedings of the 2007 international ACM conference on Supporting group work*: ACM.
- Haase, Peter, Andreas Hotho, Lars S. Thieme et York Sure. 2005. «Collaborative and Usage-Driven Evolution of Personal Ontologies.». *ESWC*. vol. 3532, p. 486-499.
- Han, Peng, Bo Xie, Fan Yang et Ruimin Sheng. 2004. «A scalable P2P recommender system based on distributed collaborative filtering». *Expert systems with applications*. En ligne. <http://linkinghub.elsevier.com/IHUB_downtime02.htm>.
- Herlocker, Jonathan L. 2000. «Understanding and improving automated collaborative filtering systems». University of Minnesota, 144 p.
- Herlocker, Jonathan L., Joseph A. Konstan, Al Borchers et John Riedl. 1999. «An algorithmic framework for performing collaborative filtering». *SIGIR '99: Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval*, p. 230-237. En ligne. <<http://doi.acm.org/10.1145/312624.312682>>.
- Herlocker, L. Jonathan, A. Joseph Konstan, G. Loren Terveen et T. John Riedl. 2004. «Evaluating collaborative filtering recommender systems». *ACM Trans. Inf. Syst.* vol. 22, no 1, p. 5-53.
- Hill, Will, Larry Stead, Mark Rosenstein et George Furnas. 1995. «Recommending and evaluating choices in a virtual community of use». *CHI '95: Proceedings of the SIGCHI conference on Human factors in computing systems*, p. 194-201. En ligne. <http://www.sigchi.org/chi95/proceedings/papers/wch_bdy.htm>.
- Hofmann, Thomas. 2003. «Collaborative filtering via gaussian probabilistic latent semantic analysis». *SIGIR '03: Proceedings of the 26th annual international ACM SIGIR conference on Research and development in information retrieval*, p. 259-266. En ligne. <<http://doi.acm.org/10.1145/860435.860483>>.

- , 2004. «Latent semantic models for collaborative filtering». *ACM Trans. Inf. Syst.* vol. 22, no 1, p. 89-115. En ligne. <<http://doi.acm.org/10.1145/963770.963774>>.
- Huang, Zan, Hsinchun Chen et Daniel Zeng. 2004. «Applying associative retrieval techniques to alleviate the sparsity problem in collaborative filtering». *ACM Trans. Inf. Syst.* vol. 22, no 1, p. 116-142.
- Huang, Zan, Daniel D. Zeng et Hsinchun Chen. 2004. «A unified recommendation framework based on Probabilistic Relational Models». in *Fourteenth Annual Workshop on Information Technologies and Systems (WITS)*, p. 8-13.
- Jacobsson, Mattias, Mattias Rost et Lars Erik Holmquist. 2006. «When Media Gets Wise: collaborative filtering with mobile media agents». In *Proceedings of the 11th international conference on Intelligent user interfaces*: ACM.
- Jennings, N. R. 1993. «Specification and Implementation of a Belief-Desire-Joint-Intention Architecture for Collaborative Problem Solving». *Intl. Journal of Intelligent and Cooperative Information Systems*. vol. 2, no 3, p. 289-318.
- Jin, Rong, Joyce Y. Chai et Luo Si. 2004. «An automatic weighting scheme for collaborative filtering». *SIGIR '04: Proceedings of the 27th annual international ACM SIGIR conference on Research and development in information retrieval*, p. 337-344. En ligne. <<http://doi.acm.org/10.1145/1008992.1009051>>.
- Kalogeraki, Vana, Dimitrios Gunopulos et D. Zeinalipour-Yazti. 2002. «A local search mechanism for peer-to-peer networks». *CIKM '02: Proceedings of the eleventh international conference on Information and knowledge management*, p. 300-307. En ligne. <<http://doi.acm.org/10.1145/584792.584842>>.
- Kappeler, Peter M. 2005. *Cooperation in Primates and Humans: Mechanisms and Evolution*: Springer.

- Kim, Jae Kyeong , Hyea Kyeong Kim et Yoon Ho Cho. 2008. «A user-oriented contents recommendation system in peer-to-peer architecture». *Expert Syst. Appl.* vol. 34, no 1, p. 300-312.
- Kinater, M., et K. Rothermel. 2003. «Architecture and Algorithms for a Distributed Reputation System». *Proceedings of the First International Conference on Trust Management.* vol. 2692, p. 1-16.
- Kobsa, Alfred 2007. «Privacy-enhanced personalization». *Commun. ACM.* vol. 50, no 8, p. 24-33.
- Kong, J. S., B. A. Rezaei, N. Sarshar, V. P. Roychowdhury et P. O. Boykin. 2006. «Collaborative Spam Filtering Using E-Mail Networks». *Computer.* vol. 39, no 8, p. 67-73.
- Konstan, Joseph A. 2004. «Introduction to recommender systems: Algorithms and Evaluation». *ACM Trans. Inf. Syst.* vol. 22, no 1, p. 1-4.
- Kumar, S. Ravi, Prabhakar Raghavan, Sridhar Rajagopalan et Andrew Tomkins. 1998. «Recommendation Systems: A Probabilistic Analysis». *IEEE Symposium on Foundations of Computer Science*, p. 664-673.
- Lam, Shyong K., Dan Frankowski et T. John Riedl. 2006. «Do You Trust Your Recommendations? An Exploration of Security and Privacy Issues in Recommender Systems». In *Emerging Trends in Information and Communication Security*, p. 14-29. En ligne. <http://dx.doi.org/10.1007/11766155_2>.
- Lam, Shyong K., et T. John Riedl. 2005. «Privacy, Shilling, and The Value of Information in Recommender Systems». *Proceedings of User Modeling Workshop on Privacy-Enhanced Personalization*, p. 85-92.

- Lang, Ken. 1995. «NewsWeeder: learning to filter netnews». *Proceedings of the 12th International Conference on Machine Learning*, p. 331-339.
- Lathia, Neal, Stephen Hailes et Licia Capra. 2007. «Private distributed collaborative filtering using estimated concordance measures». In *Proceedings of the 2007 ACM conference on Recommender systems*: ACM.
- Lemire, Daniel. 2005. «Scale and Translation Invariant Collaborative Filtering Systems». *Information Retrieval*. vol. 8, no 1, p. 129-150. En ligne. <<http://portal.acm.org/citation.cfm?id=1034894.1034899>>.
- Lemire, Daniel, et Anna Maclachlan. 2005. «Slope One Predictors for Online Rating-Based Collaborative Filtering». In *SIAM Data Mining (SDM'05)*. En ligne. <<http://cogprints.org/4031/>>.
- Leung, Cane Wing-ki, Stephen Chi-fai Chan et Fu-lai Chung. 2007. «Applying Cross-Level Association Rule Mining to Cold-Start Recommendations». *WI-IATW '07: Proceedings of the 2007 IEEE/WIC/ACM International Conferences on Web Intelligence and Intelligent Agent Technology - Workshops*, p. 133-136.
- Li, Qing, et Byeong Man Kim. 2003. «An approach for combining content-based and collaborative filters». *Proceedings of the sixth international workshop on Information retrieval with Asian languages*, p. 17-24. En ligne. <<http://dx.doi.org/10.3115/1118935.1118938>>.
- Liang, J., R. Kumar et K. Ross. 2004. «Understanding KaZaA».
- Linden, G., B. Smith et J. York. 2003. «Amazon.com recommendations: item-to-item collaborative filtering». *Internet Computing, IEEE*. vol. 7, no 1, p. 76-80.

- Liu, Duen-Ren, et Ya-Yueh Shih. 2005. «Hybrid approaches to product recommendation based on customer lifetime value and purchase preferences». *J. Syst. Softw.* vol. 77, no 2, p. 181-191.
- Lops, Pasquale, Marco Degemmis et Giovanni Semeraro. 2007. «Improving Social Filtering Techniques Through WordNet-Based User Profiles». In *Proceedings of the 11th international conference on User Modeling*: Springer-Verlag.
- Lorini, Emiliano, Nicolas Troquard, Andreas Herzig et Cristiano Castelfranchi. 2007. «Delegation and mental states». In *Proceedings of the 6th international joint conference on Autonomous agents and multiagent systems*: ACM.
- Lua, Keong, J. Crowcroft, M. Pias, R. Sharma et S. Lim. 2005. «A survey and comparison of peer-to-peer overlay network schemes». *Communications Surveys & Tutorials, IEEE*, p. 72-93.
- Lv, Qin, Pei Cao, Edith Cohen, Kai Li et Scott Shenker. 2002. «Search and replication in unstructured peer-to-peer networks». *ICS '02: Proceedings of the 16th international conference on Supercomputing*, p. 84-95. En ligne. <<http://doi.acm.org/10.1145/514191.514206>>.
- Malaisson, Claude. 2007. *Pourquoi bloguer dans un contexte d'affaires*. Montréal: Isabelle Quentin.
- Manning, Christopher D. , Prabhakar Raghavan et Hinrich Schtze. 2008. *Introduction to Information Retrieval*: Cambridge University Press. En ligne. <<http://www-csli.stanford.edu/~hinrich/information-retrieval-book.html>>.
- Marlin, Benjamin. 2003. «Modeling User Rating Profiles For Collaborative Filtering». In *Neural Information Processing Systems*.

- Massa, Paolo, et Paolo Avesani. 2004. «Trust-Aware Collaborative Filtering for Recommender Systems». *On the Move to Meaningful Internet Systems 2004: CoopIS, DOA, and ODBASE: OTM Confederated International Conferences, CoopIS, DOA, and ODBASE*. vol. 3290.
- , 2007a. «Trust-aware recommender systems». In *Proceedings of the 2007 ACM conference on Recommender systems*: ACM.
- , 2007b. «Trust metrics on controversial users: balancing between tyranny of the majority and echo chambers». *International Journal on Semantic Web and Information Systems*.
- McNee, Sean M., John Riedl et Joseph A. Konstan. 2006. «Being accurate is not enough: how accuracy metrics have hurt recommender systems». *CHI '06: CHI '06 extended abstracts on Human factors in computing systems*, p. 1097-1101. En ligne. <<http://doi.acm.org/10.1145/1125451.1125659>>.
- Mehta, Bhaskar, Claudia Niederée, Avare Stewart, Marco Degemmis, Pasquale Lops et Giovanni Semeraro (2005). Ontologically-Enriched Unified User Modeling for Cross-System Personalization. *User Modeling*: 119-123 p
- Melville, Prem, Raymod J. Mooney et Ramadass Nagarajan. 2002. «Content-boosted collaborative filtering for improved recommendations». In *Eighteenth national conference on Artificial intelligence*: American Association for Artificial Intelligence.
- Memmi, Daniel, et Olivier Nérot. 2003. «Building virtual communities for information retrieval». In *CRIWG*: Springer.
- Middleton, Stuart E., Nigel R. Shadbolt et David C. De Roure. 2004. «Ontological user profiling in recommender systems». *ACM Trans. Inf. Syst.* vol. 22, no 1, p. 54-88.

- Miller, Bradley N., Joseph A. Konstan et John T. Riedl. 2004. «PocketLens: Toward a personal recommender system». *ACM Trans. Inf. Syst.* vol. 22, no 3, p. 437-476.
- Miller, N. Bradley 2003. «Toward a personal recommender system». University of Minnesota, 185 p.
- Minsky, Marvin Lee. 1986. *The society of mind*. New York: Simon and Schuster.
- Mobasher, Bamshad, Xin Jin et Yanzan Zhou. 2004. «Semantically Enhanced Collaborative Filtering on the Web». *Web Mining: FromWeb to SemanticWeb*, p. 57-76.
- Montaner, M. 2003. «Collaborative Recommender Agents Based on Case-Based Reasoning and Trust». Girona, Spain, Universitat de Girona, Girona.
- Mooney, R., P. Bennett et L. Roy. 1998. «Book recommending using text categorization with extracted information». In *The AAAI Workshop on Recommender Systems*: AAAI Press.
- Mooney, Raymond J., et Loriene Roy. 2000. «Content-based book recommending using learning for text categorization». *Proceedings of DL-00, 5th ACM Conference on Digital Libraries*, p. 195-204.
- Mui, Lik, Peter Szolovits et Cheewee Ang. 2001. «Collaborative sanctioning: applications in restaurant recommendations based on reputation».
- Müller, J.-P. 1998. «Vers une methodologie de conception de systemes multi-agents de resolution de probleme par emergence». In *JFIADSMA '98*.

- Nakamura, Atsuyoshi, et Naoki Abe. 1998. «Collaborative Filtering Using Weighted Majority Prediction Algorithms». *ICML '98: Proceedings of the Fifteenth International Conference on Machine Learning*, p. 395-403.
- Nakauchi, Kiyohide, Yuichi Ishikawa, Hiroyuki Morikawa et Tomonori Aoyama. 2003. «Peer-to-Peer Keyword Search Using Keyword Relationship». *CCGRID '03: Proceedings of the 3rd International Symposium on Cluster Computing and the Grid*, p. 359.
- Netcraft. 2007. «May 2007 Web Server Survey». Netcraft. En ligne. <http://news.netcraft.com/archives/2007/05/01/may_2007_web_server_survey.html>.
- Newell, Allen. 1982. «The Knowledge Level». *Artificial Intelligence*. vol. 18, no 1, p. 87-127.
- Oka, T., H. Morikawa et T. Aoyama. 2004. «Vineyard : A Collaborative Filtering Service Platform in Distributed Environment». *Proc. of the IEEE/IPSJ Symposium on Applications and the Internet Workshops*.
- Olsson, Tomas. 1998. «Decentralized Social Filtering based on Trust». *Proceedings of AAAI-98 Recommender Systems Workshop*, p. 5.
- , 2003. «Bootstrapping and decentralizing recommender systems». Licentiate Thesis, Uppsala University.
- Parunak, H. 1997. «Go to the Ant: Engineering Principles from Natural Multi-Agent Systems».

- Pavlov, Dmitry Y., et David M. Pennock. 2002. «A maximum entropy approach to collaborative filtering in dynamic, sparse, high-dimensional domains». In *Neural Information Processing Systems*.
- Pazzani, Michael J. 1999. «A Framework for Collaborative, Content-Based and Demographic Filtering». *Artif. Intell. Rev.* vol. 13, no 5-6, p. 393-408.
- Pazzani, Michael J., et Daniel Billsus. 1997. «Learning and Revising User Profiles: The Identification of Interesting Web Sites». *Machine Learning*. vol. 27, no 3, p. 313-331.
- Pennock, David, Eric Horvitz, Steve Lawrence et C. Lee Giles. 2000. «Collaborative Filtering by Personality Diagnosis: A Hybrid Memory- and Model-based Approach». *Proceedings of the 16th Conference on Uncertainty in Artificial Intelligence, UAI 2000*, p. 473-480.
- Picault, S. 1998. «A Multi-Agent Simulation of Primate Social Concepts». *ECAI*, p. 327-328.
- Picault, Sebastien, et Anne Collinot. 1998. «Designing Social Cognition Models for Multi-Agent Systems through Simulating Primate Societies». In *Proceedings of the 3rd International Conference on Multi Agent Systems*: IEEE Computer Society.
- Pingfeng, Liu, Nie Guihua, Chen Donglin et A. Zhichao Fu Zhichao Fu. 2007. «The Knowledge Grid Based Intelligent Electronic Commerce Recommender Systems». In *Service-Oriented Computing and Applications, 2007. SOCA '07. IEEE International Conference on*.
- Popescul, Alexandrin, Lyle Ungar, David Pennock et Steve Lawrence. 2001. «Probabilistic Models for Unified Collaborative and Content-Based Recommendation in Sparse-Data Environments». *17th Conference on Uncertainty in Artificial Intelligence*, p. 437-444.

- Prakash, Vipul Ved, et Adam O'Donnell. 2005. «Fighting spam with reputation systems». *Queue*. vol. 3, no 9, p. 36-41.
- Quinqueton, J., et Y. Hamadi. 1999. *Journées Francophones d'Intelligence Artificielle Distribuée et de Système Multi-Agents (JFIADSMA99)* (La Réunion, France). Editions Hermès, 225-235 p.
- Rao, Anand S., et Michael P. Georgeff. 1991. «Modeling Rational Agents within a BDI-Architecture». *Proceedings of the 2nd International Conference on Principles of Knowledge Representation and Reasoning (KR'91)*, p. 473-484.
- Resnick, Mitchel. 1997. *Turtles, Termites, and Traffic Jams: Explorations in Massively Parallel Microworlds (Complex Adaptive Systems)*: The MIT Press.
- Resnick, Paul, Neophytos Iacovou, Mitesh Suchak, Peter Bergstrom et John Riedl. 1994. «GroupLens: an open architecture for collaborative filtering of netnews». *CSCW '94: Proceedings of the 1994 ACM conference on Computer supported cooperative work*, p. 175-186. En ligne. <<http://doi.acm.org/10.1145/192844.192905>>.
- Rocchio, J.J Jr. 1971. «Relevance feedback in information retrieval». *The SMART Retrieval System: Experiments in Automatic Document Processing*, p. 313-323.
- Rojsattarat, Ekkawut, et Nuanwan Soonthornphisaj. 2003. «Hybrid Recommendation: Combining Content-Based Prediction and Collaborative Filtering.». *INTELLIGENT DATA ENGINEERING AND AUTOMATED LEARNING: IDEAL*. vol. 2690, p. 337-344.
- Russell, Stuart J., et Peter Norvig. 2004. *Artificial intelligence : a modern approach*, 2nd. Coll. «Prentice Hall series in artificial intelligence». Upper Saddle River, N.J.: Prentice Hall/Pearson Education. En ligne. <<http://aima.cs.berkeley.edu/>>.

- Salton, G., A. Wong et C. S. Yang. 1975. «A vector space model for automatic indexing». *Commun. ACM*. vol. 18, no 11, p. 613-620.
- Salton, Gerard 1989. *Automatic text processing: the transformation, analysis, and retrieval of information by computer*: Addison-Wesley Longman Publishing Co., Inc.
- Sarwar, Badrul, George Karypis, Joseph Konstan et John Reidl. 2001. «Item-based collaborative filtering recommendation algorithms». In *Proceedings of the 10th international conference on World Wide Web*: ACM.
- Sarwar, Badrul, George Karypis, Joseph Konstan et T. John Riedl. 2000a. «Analysis of recommendation algorithms for e-commerce». In *Proceedings of the 2nd ACM conference on Electronic commerce*: ACM.
- , 2000b. «Application of dimensionality reduction in recommender systems-a case study». In *ACM WebKDD 2000 Web Mining for E-Commerce Workshop*.
- Schein, A., A. Popescul, L. Ungar et D. Pennock. 2002. «Methods and metrics for cold-start recommendations». *Proceedings of the 25th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 2002)*, p. 253-260.
- Schein, Andrew I., Alexandrin Popescul, Lyle H. Ungar et David M. Pennock. 2005. «CROC: A New Evaluation Criterion for Recommender Systems». *Electronic Commerce Research*. vol. 5, no 1, p. 51-74.
- Schickel-Zuber, Vincent, et Boi Faltings. 2006. «Inferring User's Preferences using Ontologies». In *AAAI 2006* (2006). En ligne. <<http://www.aaai.org/Library/AAAI/aaai06contents.php>>.

- Searle, John R. 1969. *Speech acts an essay in the philosophy of language*. Cambridge, Angleterre: Cambridge University Press.
- Shani, Guy, Ronen I. Brafman et David Heckerman. 2002. «An MDP-Based Recommender System». In *18th Conf. Uncertainty in Artificial Intelligence*.
- Shardanand, Upendra, et Patti Maes. 1995. «Social Information Filtering: Algorithms for Automating ``Word of Mouth"». *Proceedings of ACM CHI'95 Conference on Human Factors in Computing Systems*. vol. 1, p. 210-217.
- Si, L., et R. Jin. 2003. «A Flexible Mixture Model for Collaborative Filtering». In *20th Int'l Conf. Machine Learning*.
- Soboroff, Ian, et Charles Nicholas. 1999. «Combining Content and Collaboration in Text Filtering». *Proceedings of the IJCAI-99 Workshop on Machine Learning for Information Filtering*.
- Steels, Luc. 2000. «Language as a Complex Adaptive System». *Proceedings of PPSN VI*.
- , 2003a. «Evolving grounded communication for robots». *Trends in Cognitive Sciences*. vol. 7, no 7, p. 308-312. En ligne. <10.1016/S1364-6613(03)00129-3>.
- , 2003b. «Intelligence with representation». *Philosophical Transactions: Mathematical, Physical and Engineering Sciences*. vol. 361, no 1811, p. 2381-2395. En ligne. <10.1098/rsta.2003.1257>.
- Stoica, Ion, Robert Morris, David Karger, Frans F. Kaashoek et Hari Balakrishnan. 2001. «Chord: A scalable peer-to-peer lookup service for internet applications». *SIGCOMM Comput. Commun. Rev.* vol. 31, no 4, p. 149-160. En ligne. <10.1145/964723.383071>.

- Stoti, Nadia (2006). Inférer la confiance dans les systèmes de recommandation collaboratifs. Cours : gestion des connaissances. Montreal, HEC
- Stritt, Manuel, Karen Tso et Lars Schmidt-Thieme. 2006. «Attribute-Aware Anonymous Recommender Systems». In *30th Annual Conference of the Gesellschaft für Klassifikation (GfKI)*.
- Sullivan, Danny. 2006. «Searches Per Day». Search Engine Watch. En ligne. <<http://searchenginewatch.com/showPage.html?page=2156461>>.
- Szomszor, Martin, Ciro Cattuto, Harith Alani, Kieron O'äöhara, Andrea Baldassarri, Vittorio Loreto et Vito D. P. Servedio (2007). Folksonomies, the Semantic Web, and Movie Recommendation. Bridging the Gap between Semantic Web and Web 2.0 (SemNet 2007): 71-84 p
- Terveen, G. Loren, et Will Hill. 2001. «Beyond Recommender Systems: Helping People Help Each Other». In *Human-Computer Interaction in the New Millenium*, John M. Carroll. Reading, MA, USA.
- Tintarev, N., et J. Masthoff. 2007. «A Survey of Explanations in Recommender Systems». In *Data Engineering Workshop, 2007 IEEE 23rd International Conference on*.
- Tran, T., et R. Cohen. 2000. «Hybrid Recommender Systems for Electronic Commerce». *Knowledge-Based Electronic Markets, Papers from the AAAI Workshop, AAAI Technical Report WS-00-04*, p. 78-83.
- Tso-Sutter, Karen H. L., Leandro Balby Marinho et Lars Schmidt-Thieme. 2008. «Tag-aware recommender systems by fusion of collaborative filtering algorithms». In *Proceedings of the 2008 ACM symposium on Applied computing*: ACM.

- Tsoumakos, Dimitrios, et Nick Roussopoulos. 2003. «Adaptive Probabilistic Search for Peer-to-Peer Networks». *P2P '03: Proceedings of the 3rd International Conference on Peer-to-Peer Computing*, p. 102.
- Ujjin, Supiya, et Peter J. Bentley. 2001. «Building a Lifestyle Recommender System». In *The Tenth International World-Wide-Web Conference*. En ligne. <<http://wwwconf.ecs.soton.ac.uk/archive/00000086/>>.
- Ungar, L., et D. Foster. 1998. «Clustering Methods For Collaborative Filtering». *Proceedings of the Workshop on Recommendation Systems*.
- Vaario, Jari. 1994. «Modeling Biological Adaptation». *Japan-U.S.A. Symposium on Flexible Automation, Bionic Systems and Artificial Life*, July 11-18.
- Van Setten, Mark, Mettina Veenstra, Anton Nijholt et Betsy Van Dijk. 2004. «Case-based reasoning as a prediction strategy for hybrid recommender systems». *Second International Atlantic Web Intelligence Conference (AWIC 2004)*. vol. 3034, p. 13-22.
- Vellino, Andre, et David Zeber. 2007. «A Hybrid, Multi-dimensional Recommender for Journal Articles in a Scientific Digital Library». *WI-LATW '07: Proceedings of the 2007 IEEE/WIC/ACM International Conferences on Web Intelligence and Intelligent Agent Technology - Workshops*, p. 111-114.
- (2008). *Recommending Journal Articles in a Scientific Digital Library*. Montreal, École de technologie supérieure
- Visetti, Yves-Marie. 2004. «Constructivismes, émergences : une analyse sémantique et thématique ». *Intellectica : Des lois de la pensée aux constructivismes*. vol. 39, no 2, p. 229-259.

- Wang, Jun, Arjen P. de Vries et Marcel J. T. Reinders. 2006. «Unifying user-based and item-based collaborative filtering approaches by similarity fusion». *SIGIR '06: Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval*, p. 501-508. En ligne. <<http://doi.acm.org/10.1145/1148170.1148257>>.
- , 2008. «Unified relevance models for rating prediction in collaborative filtering». *ACM Trans. Inf. Syst.* vol. 26, no 3, p. 1-42. En ligne. <<http://doi.acm.org/10.1145/1361684.1361689>>.
- Wang, Jun, Johan Pouwelse, L. Reginald Lagendijk et J. T. Marcel Reinders. 2006a. «Distributed collaborative filtering for peer-to-peer file sharing systems». In *Proceedings of the 2006 ACM symposium on Applied computing*: ACM.
- Wang, Jun, Johan Pouwelse, Jenneke Fokker et Marcel J.T. Reinders. 2006b. «Personalization of a Peer-to-Peer Television System». *Proc. of European Conference on Interactive Television (EuroITV 2006)*.
- Widyantoro, Dwi H., Thomas R. Loerger et John Yen. 2001. «Learning user interest dynamics with a three-descriptor representation». *Journal of the American Society of Information Science*. vol. 52, no 3, p. 212-225. En ligne. <[http://dx.doi.org/10.1002/1532-2890\(2000\)9999:9999<::AID-AS11615>3.3.CO;2-F](http://dx.doi.org/10.1002/1532-2890(2000)9999:9999<::AID-AS11615>3.3.CO;2-F)>.
- Wittner, Otto, Poul E. Heegaard et Bjarne E. Helvik (2003). *Swarm Based Distributed Search in the AMIGOS Environment*, Department of Telematics, Norwegian University of Science and Technology
- Xue, Gui-Rong, Chenxi Lin, Qiang Yang, WenSi Xi, Hua-Jun Zeng, Yong Yu et Zheng Chen. 2005. «Scalable collaborative filtering using cluster-based smoothing». *SIGIR '05: Proceedings of the 28th annual international ACM SIGIR conference on Research and development in information retrieval*, p. 114-121. En ligne. <<http://doi.acm.org/10.1145/1076034.1076056>>.

- Yang, Beverly, et Hector Garcia-Molina. 2001. «Comparing Hybrid Peer-to-Peer Systems». *VLDB '01: Proceedings of the 27th International Conference on Very Large Data Bases*, p. 561-570.
- Yu, Kai, Anton Schwaighofer et Volker Tresp. 2003. «Collaborative ensemble learning: Combining collaborative and content-based information filtering via hierarchical bayes». In *Proceedings of the 19th Conference on Uncertainty in Artificial Intelligence*, p. 616-623.
- Yu, Kai, Anton Schwaighofer, Volker Tresp, Xiaowei Xu et Hans-Peter Kriegel. 2004. «Probabilistic Memory-Based Collaborative Filtering». *IEEE Trans. on Knowl. and Data Eng.* vol. 16, no 1, p. 56-69. En ligne. <<http://dx.doi.org/10.1109/TKDE.2004.1264822>>.
- Yu, Kai, Xiaowei Xu, Jianjua Tao, Martin Ester et Hans-Peter Kriegel. 2002. «Instance Selection Techniques for Memory-based Collaborative Filtering». In *Second SIAM Int'l Conf. Data Mining (SDM '02)*.
- Zaier, Zied. 2002. «Apport de l'intelligence artificielle dans les outils de recherche d'informations». Mémoire, Montréal, Université du Québec à Montréal, xiii, 165 f. p.
- , 2008a. «Finalement un système de recommandation pour flux RSS.». En ligne. <<http://ziedzaier.com/nouvelles/finalement-un-systeme-de-recommandation-pour-flux-rss.html>>.
- , 2008b. «Torture les données, jusqu'à ce qu'elles confessent.». En ligne. <<http://ziedzaier.com/nouvelles/torture-les-donnees-jusqu%E2%80%99a-ce-qu%E2%80%99elles-confessent.html>>.
- Zaier, Zied, Robert Godin et Luc Faucher. 2007. «Recommendation Quality Evolution Based on Neighborhood Size». In *Third International Conference on Automated Production of Cross Media Content for Multi-Channel Distribution. AXMEDIS '07*.

- , 2008a. «Evaluating Recommender Systems». In *Fourth International Conference on Automated Production of Cross Media Content for Multi-Channel Distribution. AXMEDIS '08*.
- , 2008b. «Recommendation Quality Evolution Based on Neighbors Discrimination». In *MCETECH Conference on e-Technologies*.
- Zeinalipour-Yazti, Demetrios, Vana Kalogeraki et Dimitrios Gunopulos. 2004. «Information Retrieval Techniques for Peer-to-Peer Networks». *Computing in Science and Engg.* vol. 6, no 4, p. 20-26. En ligne. <<http://dx.doi.org/10.1109/MCSE.2004.12>>.
- Zenebe, Azene, et Anthony F. Norcio. 2005. «Evaluation Framework for Fuzzy Theoretic-Based Recommender System ». In *HCI International 2005: The 11th International Conference on Human-Computer Interaction*.
- Zhang, Yi, Jamie Callan et Thomas Minka. 2002. «Novelty and redundancy detection in adaptive filtering». *SIGIR '02: Proceedings of the 25th annual international ACM SIGIR conference on Research and development in information retrieval*, p. 81-88. En ligne. <<http://doi.acm.org/10.1145/564376.564393>>.
- Ziegler, C., L. Schmidt-Thieme et G. Lausen (2004). Exploiting Semantic Product Descriptions for Recommender Systems. Proc. of the 2nd ACM SIGIR Semantic Web and Information Retrieval Workshop (SWIR '04). Sheffield, UK
- Ziegler, Cai-Nicolas. 2005. «Towards Decentralized Recommender Systems». Albert-Ludwigs-Universitat Freiburg - Fakultat fur Angewandte Wissenschaften, Institut fur Informatik.
- Ziegler, Cai-Nicolas, et Jennifer Golbeck. 2007. «Investigating interactions of trust and interest similarity». *Decis. Support Syst.* vol. 43, no 2, p. 460-475.

Ziegler, Cai-Nicolas, Sean M. McNee, Joseph A. Konstan et Georg Lausen. 2005.
«Improving recommendation lists through topic diversification». In *Proceedings of
the 14th international conference on World Wide Web*: ACM.